

Cedric F. Jacob

Mastering Zendesk

Master the art of providing effective IT services to your customers by leveraging Zendesk



Packt

Mastering Zendesk

Master the art of providing effective IT services to your customers by leveraging Zendesk

Cedric F. Jacob

Packt

BIRMINGHAM - MUMBAI

Mastering Zendesk

Copyright © 2017 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: January 2017

Production reference: 1100117

Published by Packt Publishing Ltd.

Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78646-104-9

www.packtpub.com

Credits

Author

Cedric F. Jacob

Copy Editor

Tom Jacob

Reviewer

Maria Herrmann

Project Coordinator

Kinjal Bari

Commissioning Editor

Pratik Shah

Proofreader

Safis Editing

Acquisition Editor

Prachi Bisht

Indexer

Francy Puthiry

Content Development Editors

Sanjeet Rao

Monika Sangwan

Graphics

Kirk D'Penha

Technical Editors

Bhagyashree Rai

Nidhisha Shetty

Production Coordinator

Shantanu Zagade

About the Author

Cedric F. Jacob, before starting his own business as a customer service consultant, writer, and Zendesk admin for international companies, began his career as a customer service representative, from where he progressed to senior, team leader, and management roles. Working for clients such as Runtastic and Lovoo, he focuses on creating highly customized, scalable structures, utilizing his acquired customer service experience and broader passion for IT, which ranges from software and game development all the way to cyber security and hardware.

I would like to thank Sanjeet Rao and Prachi Bisht from Packt publishing, who trusted me with this project and guided me through the process of writing my first book.

I would also like to thank everyone who joined me on my professional journey so far and helped me to achieve my goals, including Maria Herrmann, who graciously accepted to review this book. Many thanks also go to (in no particular order): Philipp Theißler, Christian Thaller, Johannes Strassmann, Claudio Di Vincenzo, Pauline Gehrmann-Golomer, and Max Uhlmann.

A special thank you to my family, who always supported me in my ambitions. Just so I can include your names in this book and ordered by age, thank you Hiltrud, Francis, Denise, and Laura.

Lastly and importantly, I would like to thank my girlfriend Stéphanie, who kept me company through long nights of writing, proofread my first drafts, and always made sure that I would not run out of caffeine during the process. Thank you!

About the Reviewer

Maria Herrmann holds a Masters degree in Language and Communication, leading her to work as an online editor, technical translator, and communication strategist in companies with an international background. For the last few years she has been working in the customer service department of an international technology start-up in Germany – setting up, administrating, and troubleshooting a multilingual Zendesk environment, including: processing large amounts of incoming tickets by using complex triggers and automations for several channels, creating custom macros and applications, producing statistics to improve performance, and designing and implementing a help center.

Firstly, my appreciation goes to the author himself, with whom I have worked on numerous tasks and projects. Discussing, implementing, and finally seeing ideas come to life in Zendesk is a very fulfilling process and I am convinced that this book will help the reader achieve this goal exactly. Secondly, I am more than grateful to my husband and daughter who made it possible for me to dedicate the necessary time to review this book – thus helping to make it as good as possible for the reader.

www.PacktPub.com

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www.packtpub.com/mapt>

Get the most in-demand software skills with Mapt. Mapt gives you full access to all Packt books and video courses, as well as industry-leading tools to help you plan your personal development and advance your career.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Customer Feedback

Thank you for purchasing this Packt book. We take our commitment to improving our content and products to meet your needs seriously—that's why your feedback is so valuable. Whatever your feelings about your purchase, please consider leaving a review on this book's Amazon page. Not only will this help us, more importantly it will also help others in the community to make an informed decision about the resources that they invest in to learn. You can also review for us on a regular basis by joining our reviewers' club. **If you're interested in joining, or would like to learn more about the benefits we offer, please contact us:** customerreviews@packtpub.com.

Table of Contents

Preface	1
Chapter 1: Configuring Your Own Zendesk	7
The Zendesk environment	8
The basic Zendesk setup	10
Channels	10
Self-service	11
Apps and integrations	12
Ticketing workflows and efficiency	12
Triggers and automations	12
Views	13
Macros	13
Service Level Agreements	14
Groups	14
Organizations	14
Performance reporting	14
Localization and branding	15
Evaluating individual requirements	16
Our example company	16
Desired setup	17
The initial contact/channels	17
Email	18
Twitter	18
Facebook	18
Widget	18
Help Center / Support Form	18
The ticket's journey	18
Ticket views	20
Agent workflow	22
Zendesk setup	25
Asking questions	25
Listing tasks	29
Creating the road map	30
Summary	31
Chapter 2: Agent Roles, Groups, Organizations, and User Tags	32
Users/agents	33
Custom agent roles	34
Ticket permissions	36

People permissions	39
Help Center permissions	40
Tools permissions	40
Channels permissions	42
System permissions	43
Custom roles for our example	43
Groups	44
Organizations	47
Tags	50
Domains	50
Group	50
Users	51
Details	51
Notes	51
Organizations for our example	51
User tags	52
Importing existing user databases	54
Summary	57
Chapter 3: Creating Custom Fields	58
What are fields in Zendesk?	58
Standard ticket fields	59
Requester	60
Assignee	60
CCs	60
Share	60
Subject	60
Description	60
Status	61
Type	61
Priority	61
Tags	62
Standard user and organization fields	62
Different types of fields	63
Drop-down list	64
Text	64
Multi-line text	65
Numeric	65
Decimal	65
Checkbox	66
Date	66
Regular Expression	66
Creating custom fields	67
Custom user fields	68

Custom ticket fields	71
Custom organization fields	75
Summary	76
Chapter 4: Setting Up Multiple Ticket Channels	77
What are channels?	78
Email channel	78
Facebook channel	80
Twitter channel	80
Chat (Zopim) channel	81
Talk channel	82
Web Widget channel	83
Mobile SDK channel	84
API channel	84
Setting up Zendesk channels	86
Email channel setup	86
Adding internal e-mail addresses	86
Forwarding example with Outlook.com	89
Create SPF record	92
Email channel settings	94
Accept wildcard emails	94
Personalized email replies	94
Gmail Go-to Actions	94
Email templates	95
Mail delimiter	96
Custom domain for DKIM	96
Facebook channel setup	97
Adding your first Facebook page	97
Facebook channel settings	99
Twitter channel setup	100
Adding your first Twitter account	100
Twitter channel settings	101
Account-related settings	102
General settings	103
Manage searches	104
Widget channel setup	106
About Zendesk's widget	106
Widget customization	106
Widget snippet code	110
Help Centre (Support Form) channel setup	111
Zendesk's Help Centre— A quick overview	111
The support form	115
Summary	117

Chapter 5: Customizing Business Rules and Ticket Escalation

118

What are Business Rules?	119
Triggers versus automations	120
Triggers in detail	121
Default triggers	123
Notify requester of received request	124
Notify requester of comment update	128
Notify assignee of comment update	130
Notify assignee of assignment	133
Notify assignee of reopened ticket	135
Notify group of assignment	136
Notify all agents of received request	139
Default trigger conclusion	140
Conditions	141
Ticket-related conditions	141
Ticket: Status	142
Ticket: Type	142
Ticket: Priority	143
Ticket: Group	144
Ticket: Assignee	144
Ticket: Requester	145
Ticket: Organization	146
Ticket: Tags	146
Ticket: Channel	147
Ticket: Update via	148
Ticket: Received at	148
Ticket: Satisfaction	148
Ticket: Is...	149
Ticket: Comment is...	150
Ticket: Comment text...	150
Ticket: Reopens	151
Ticket: Agent replies	151
Ticket: Custom Fields	152
Requester related conditions	152
Requester: Language	152
Requester: Role	153
Requester: Time zone	153
Requester: Custom Fields	153
Other Conditions	153
Other: Current user	154
Organization: Custom Fields	154
Actions	154
Ticket-related actions	154
Ticket: Status	155
Ticket: Priority	155
Ticket: Type	156

Ticket: Group	157
Ticket: Assignee	157
Ticket: Satisfaction	157
Ticket: Set tags	158
Ticket: Add tags	158
Ticket: Remove tags	158
Ticket: Add CC	158
Ticket: Custom Fields	159
Requester- and organization-related actions	159
Requester: Language	159
Requester: Custom fields	159
Organization: Custom fields	159
Other actions	159
Notifications: Email user	160
Notifications: Email group	161
Creating custom triggers	161
Additional tips	166
Naming and organizing triggers	166
Testing triggers	167
Automations in detail	168
Default automations	171
Close ticket 4 days after status is set to solved	171
Request customer satisfaction rating	172
Automation-specific conditions	175
Creating custom automations	177
Custom ticket views	181
Service Level Agreements	182
Summary	185
Chapter 6: Integrating and Extending Zendesk	186
Zendesk apps	186
Zendesk Marketplace	187
Zendesk Labs	191
Channels	191
Zendesk app locations	191
Zendesk App examples	193
Show Related Tickets app	193
Iframe app	199
Out of Office app	202
Creating custom Zendesk apps	203
Planning a custom app	203
Installing the necessary tools	204
Installing Ruby	204
Installing Ruby in Windows	206
Installing Ruby in Mac	206
Installing the ZAT gem	206

Creating app files	206
Previewing our app	213
Coding our app	215
Uploading/installing our app	218
JIRA integration	
Integration preparations	222
Installing the integration	228
JIRA server specifics	231
Salesforce integration	
Integration requirements	232
Installing the integration	232
Viewing Salesforce data in Zendesk	237
Adding more features	243
Summary	244
Chapter 7: Advanced Reporting and Insights via GoodData	245
Reports in Zendesk	246
Creating and exporting reports in Zendesk	246
The Zendesk's reporting overview	252
Overview	253
Leaderboard	257
Knowledge Base	258
Community	260
Search	261
Net Promoter Score	263
Satisfaction	264
Insights	266
Zendesk Insights/GoodData	267
Creating a Dashboard and Tab	270
Adding reports	272
Adding filters	276
Adding Widgets	280
Adding some text and lines	283
Creating custom reports	284
Creating custom metrics	295
Schedule dashboard E-mails	302
Summary	303
Chapter 8: Security Settings and SSO	304
Security settings and best practice	304
Access security	305
Zendesk password authentication	306

Two-factor authentication	310
Zendesk's single sign-on options	312
Social media SSO	312
Enterprise SSO	313
Secure Assertion Markup Language (SAML)	322
IP restrictions	327
Using custom roles to reduce unnecessary access	328
Auditing your Zendesk account	328
Secure coding practices	329
Removing/Redacting Credit Card numbers	329
Private attachments	330
Advanced Security Enterprise Add-on	331
Data at rest encryption	331
Enhanced disaster recovery	332
HIPAA compliance	332
In case of a breach	332
Summary	333
Chapter 9: Troubleshooting Zendesk	334
Troubleshooting performance issues	334
Zendesk status	335
General browser issues	337
Disabling extensions/plugins	337
Clearing browsing data/cache	337
Traceroute	338
Troubleshooting business rules	339
Triggers and automations	339
Ticket events	339
Trigger order	342
SLAs	343
Troubleshooting Zendesk apps	343
Contacting Zendesk support	345
Summary	345
Chapter 10: Zendesk Tips and Tricks	346
Business rules	347
Notifying the team leader about a ticket with a high amount of replies	347
Using tags to disable or enable business rules for tickets	348
Closing a ticket via a tag	349
Disabling the satisfaction survey via a tag	350
Removing tags from follow-up tickets	350
Using business rules to detect and fix user errors	352
Adding comments to tickets via a URL target	353

Logging ticket events via an e-mail target	357
Roles and views	
Creating a vacation view	359
Creating a history view	360
Creating limited custom roles for new staff members	361
Reporting	
Report tags and folders in GoodData	362
Duplicating reports in GoodData	364
Excluding specific tickets from your GoodData reports	364
Sending agent-specific reports using variables	367
Zendesk apps	
Custom tagging app	372
Custom ticket merge app	375
Custom macro statistics app	376
Custom Zendesk backup app	378
Zendesk CSS hack in apps	378
Summary	382
Index	383

Preface

Zendesk is one of the leading cloud-based customer service solutions, favored by many due to its simple yet beautiful design, ease of use, efficiency, flexibility, and low cost of ownership. Behind its simplicity lies an array of features, allowing its customers to create complex workflows for small and large businesses alike.

If you're a Zendesk administrator and are eagerly waiting to dive into advanced-level concepts, then this book is for you as it aims to explore Zendesk's palette of features, with a strong focus on customization.

We look at core functionalities such as managing users, groups, and organizations. We take a close look at creating custom fields, setting up channels and adding our own business rules. Before we go on to review Zendesk's extension and integration capabilities by connecting JIRA and Salesforce to create a larger ecosystem of tools, we take a short dive into programming our own Zendesk apps. Towards the end, we not only emphasize security and troubleshooting aspects, but also provide tips and tricks to create a more efficient support environment.

What this book covers

Chapter 1, Configuring Your Own Zendesk, covers the basic Zendesk setup before moving on to the evaluation process of individual requirements and the creation of a customization roadmap.

Chapter 2, Agent Roles, Groups, Organizations, and User Tags, covers some of the most basic Zendesk components including roles, groups, organizations, and user tags. Each component is explained and put into context for later use.

Chapter 3, Creating Custom Fields, gives a detailed look at Zendesk fields before moving on to the creation of custom fields and what they can be used for.

Chapter 4, Setting Up Multiple Ticket Channels, covers the idea behind Zendesk channels, what channels are available to us, what they can be used for, and how to set them up according to our roadmap.

Chapter 5, Customizing Business Rules and Ticket Escalation, gives a detailed look at Zendesk's business rules. It covers their structure and components, how existing business rules work, and how to create our own custom solutions.

Chapter 6, *Integrating and Extending Zendesk*, covers custom apps and how we can create them to add more functionality to our Zendesk setup. It then moves on to show in great detail how Zendesk can be integrated and extended utilizing third party tools such as JIRA and Salesforce.

Chapter 7, *Advanced Reporting and Insights via GoodData*, gives a quick overview about Zendesk's very own reporting capabilities, before moving on to the creation of more sophisticated reports utilizing GoodData.

Chapter 8, *Security Settings and SSO*, covers Zendesk's general security settings alongside best practice recommendations for practical application. It then moves on to the available Single Sign-on options and how to set them up correctly.

Chapter 9, *Troubleshooting Zendesk*, consists of a quick troubleshooting guide covering general performance issues, faulty business rules, and issues with custom apps.

Chapter 10, *Zendesk Tips and Tricks*, offers a compilation of tips and tricks regarding business rules, roles, views, reporting, and Zendesk apps. This chapter focuses on thinking outside of the box.

What you need for this book

To use this book, you will need the following:

- Internet access
- Zendesk account
- Text editor

Who this book is for

This book is for proficient Zendesk administrators who want to unlock the full potential of their Zendesk environment by gaining a deeper understanding of Zendesk's advanced customization options.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "For instance, if a user is tagged with the `vip` tag, all their tickets will subsequently be tagged with the `vip` tag as well."

A block of code is set as follows:

```
curl -v -u {email_address}:{password}
https://{{subdomain}}.zendesk.com/api/v2/users.json \
-H "Content-Type: application/json"
-X POST
-d '{"user": {"name": "FirstName LastName", "email": "user@example.org"}}'
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
curl -v -u {email_address}:{password}
https://{{subdomain}}.zendesk.com/api/v2/users.json \
-H "Content-Type: application/json"
-X POST
-d '{"user": {"name": "FirstName LastName", "email": "user@example.org"}}'
```

Any command-line input or output is written as follows:

```
gem install zendesk_apps_tools
```

New terms and important words are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Click on **People** located under **MANAGE** within the admin menu."

Warnings or important notes appear in a box like this.



Tips and tricks appear like this.



Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of. To send us general feedback, simply e-mail feedback@packtpub.com, and mention the book's title in the subject of your message. If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for this book from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

You can download the code files by following these steps:

1. Log in or register to our website using your e-mail address and password.
2. Hover the mouse pointer on the **SUPPORT** tab at the top.
3. Click on **Code Downloads & Errata**.
4. Enter the name of the book in the **Search** box.
5. Select the book for which you're looking to download the code files.
6. Choose from the drop-down menu where you purchased this book from.
7. Click on **Code Download**.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of:

- WinRAR / 7-Zip for Windows
- Zipeg / iZip / UnRarX for Mac
- 7-Zip / PeaZip for Linux

The code bundle for the book is also hosted on GitHub at <https://github.com/PacktPublishing/Mastering-Zendesk>. We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from https://www.packtpub.com/sites/default/files/downloads/MasteringZendesk_ColorImages.pdf.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books-maybe a mistake in the text or the code-we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at questions@packtpub.com, and we will do our best to address the problem.

1

Configuring Your Own Zendesk

Zendesk is definitely one of the most straightforward and easier environments to set up. Its sleek design and intuitive processes are most probably one of the reasons why so many companies started using Zendesk. It screams simplicity and simplicity is fast.

Mastering Zendesk, however, can be challenging, and acquiring the necessary knowledge ends up being a time-consuming process. Relying on loose bits of information throughout the Internet to comply with the ever-growing and more complex requirements for your support environment can lead to a lot of backpedaling and frustration.

Since you are reading this book, I can safely assume that you have been working with Zendesk for a while or at the very least, that you have already decided to work with Zendesk in the future. Either way, most likely, you are already familiar with its core functionality based on tickets and are looking forward to making the most out of your support environment.

Also very likely, your current Zendesk environment has already been customized. Therefore, this chapter will quickly go over a basic Zendesk setup, which will serve as the base for all the upcoming changes throughout this book. This chapter will help you evaluate your individual requirements, planning the desired workflows as well as creating a road map for the final implementations by example.

At the end of this chapter, you will have refreshed your memory of a basic setup. You will have gained an understanding of Zendesk's customization capabilities and how to plan the final implementation.

This chapter will cover the following topics:

- A quick overview – The Zendesk environment
- The basic Zendesk setup and its components
- Evaluating individual requirements by example
- Creating a road map for your customization

The Zendesk environment

Before we go through the basic Zendesk setup, let's take a quick look at our Zendesk environment.

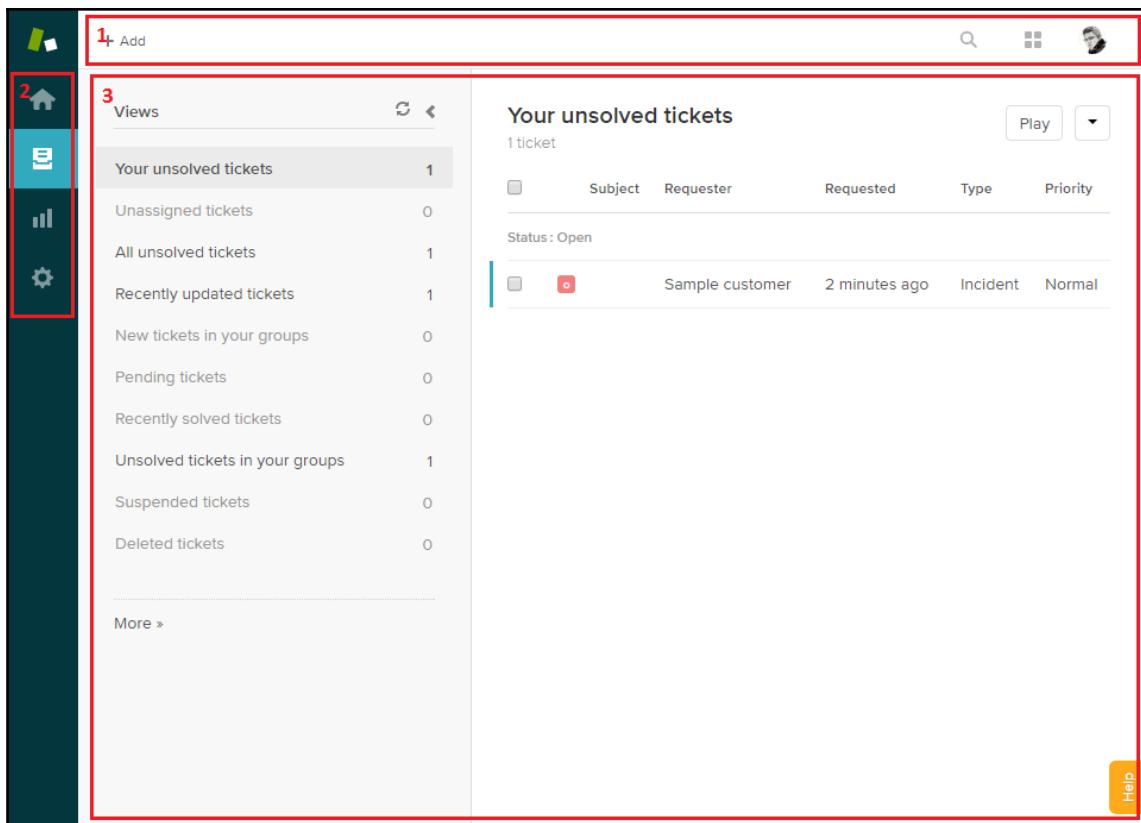
Zendesk is a customer service platform based on tickets. Customers can create tickets in order to receive help from a support agent. The support agent will then be able to review the ticket, engage in a message-based conversation with the customer, and finally set the ticket to be solved.

While this is very much an oversimplified breakdown of what Zendesk does, it serves the purpose of introduction for those who have never heard about Zendesk.

Let's take a quick look at the actual Zendesk environment and focus on its individual elements:

- Top bar
- Sidebar

- Main area



The **Top bar** can be divided into two parts. The left side, the first part, consists of ticket tabs. Zendesk allows you to work on multiple tickets simultaneously. Each ticket, similar to the pages in a web browser, is displayed as single tab. The right side, the second part, consists of a few buttons, which allow us to search our Zendesk for navigating to our Help Center or to sign out of our Zendesk environment.

The **Sidebar** consists of buttons that give us the option to navigate through Zendesk. It allows us to display our ticket views, open our individual dashboard, review the latest reports, as well as navigate to all the Zendesk settings.

Both the Top bar and the Sidebar can also display buttons to open other Zendesk apps that we can choose to add to our setup.

The **Main** area displays whatever environment we choose to open, such as our individual dashboard, Zendesk settings, or as in our case here – our ticket views. This is where most agents will open one of the views in order to pick the next ticket.

As we can see, on the surface Zendesk seems very straightforward to use. This is one of the reasons, no question about it, why agents find working with Zendesk on a daily basis so easy. However, like most systems, under the hood, there is a lot more going on than meets the eye at the first glance. This becomes more apparent when looking at a basic Zendesk setup.

The basic Zendesk setup

Once you have created a new account, Zendesk will present you a button labeled **Get Started**. If clicked, it will guide you through the following steps:

1. Set up email.
2. Invite your team.
3. Set up Help Center.
4. What's next?

This is meant to give you a quick overview of Zendesk's core functionality and point you into the right direction for further customization. While the first few steps are of little significance to us, step 4 provides us with a great overview over the basic elements of Zendesk. We will use this overview to quickly cover the basics and to freshen up our memory about the basic Zendesk setup.

Channels

Channels allow customers to contact us using different methods of communication.

Before adding any extra channels, the standard setup will allow users to create tickets via e-mail only. Out of the box, this e-mail will be `support@yourcompany.zendesk.com`.

Zendesk allows us to add a variety of channels, such as the following:

- Email
- Voice
- Chat

Channels
Channels are the ways in which your customers engage with you.

Email Voice Chat Twitter

Facebook

Self-service

Zendesk also allows us to set up self-service for our customers. The **Help Center**, for instance, can be a great tool in order to reduce customer queries by providing an open knowledge base for our customers.

All the **Self-Service** options have to be set up manually before they can be used:

Self-Service
Knowledge base, communities, and widgets to empower your customers.

Help Center Web Widget Mobile SDK

Apps and integrations

Apps are a great way to add functionality to your Zendesk environment. You may choose one of the many apps available or decide to program your own and create more individual solutions.

While Zendesk is a powerful tool, for most companies, it is likely one among a few tools within their ecosystem. Having the option to integrate Zendesk within that ecosystem can be very helpful.

Some of the more common tools Zendesk allows to be integrated are:

- JIRA
- Salesforce
- MailChimp

Ticketing workflows and efficiency

With the ticket system being at the core of Zendesk, it comes as no surprise that a great range of options are provided when it comes to tickets and the different workflows associated with them. The following elements can be used to create efficient workflows:

- Triggers and automations
- Views
- Macros
- Service Level Agreements
- Groups
- Organizations

Triggers and automations

Triggers and automations are part of the Zendesk business rules. While triggers take action when a ticket has been updated or created, automations take action after a specified amount of time. Out of the box, there are seven active triggers and one active automation.

Serving as examples, they also cover the most basic business rules for a functioning setup:

Active triggers (7)	
	Notify requester of received request
	Notify requester of comment update
	Notify assignee of comment update
	Notify assignee of assignment
	Notify assignee of reopened ticket
	Notify group of assignment
	Notify all agents of received request
Active automations (1)	
	Close ticket 4 days after status is set to solved

Views

Views are a pool of tickets, filtered by a list of set criteria. They are a crucial part when it comes to managing your ticket workflow. For example, if you offer different levels of support, views can be used to divide your tickets in Tier 1 and Tier 2 support.

Macros

A macro is a templated response for agents, that can be used to answer customers. Mainly used to ensure the same level of support by providing macros for your agents, macros are more than just text modules. Creating macros, you can choose specific ticket properties that you would like to change upon use. Good examples are, adding a predefined tag, changing the status of a ticket, and setting custom fields whenever an agent uses a macro.

Service Level Agreements

A **Service Level Agreement (SLA)** can be seen as a contract between you as a service provider and your customers, outlining the level of service that can be expected in a certain time frame. This feature is also helpful when using reports to review your team's performance.

Groups

Groups can come very handy when defining and implementing workflows. For instance, a trigger could check a ticket for the keyword “lawyer”. If the keyword is found, the ticket is pushed to the group “supervisors”. Now, only agents in that group can access that specific ticket.

Organizations

Each user (including end users) can be assigned to an organization. This can be useful in a handful of different scenarios such as restricting certain content in your help center to only one organization. Like a lot of Zendesk's options, you may come up with your own purpose of use.

Zendesk categorizes all the preceding elements under the **Ticketing Workflows and Efficiency** heading:

Ticketing Workflows and Efficiency
Optimize your support workflows and improve agent performance.

 Triggers	 Automations			
 Views	 Macros	 Service Level Agreements	 Groups	 Organizations

Performance reporting

While Zendesk provides options for performance reports, most Zendesk admins make use of the GoodData integration. GoodData is a powerful reporting and analytics tool working with Zendesk right out of the box.

If the provided reporting options are sufficient for you, Zendesk allows you to get detailed snapshots in different areas of support, as shown in the following screenshot:



Localization and branding

Being able to offer support in more than just one language is important for most international companies. Hence, it comes as no surprise that Zendesk also offers the necessary options to do so.

Most importantly, Zendesk allows you to create dynamic content, which can be referenced by a placeholder within your macros or business rules. A placeholder would look something like this:

```
{ {dc.example_placeholder} }
```

Using dynamic content means that Zendesk will automatically display the right language text depending on the user's language settings.

Zendesk classes this feature as part of their localization options:





It is important to note that Zendesk does not guide you through every possible setting. While they do cover the basics, there are many options hidden within the **Settings** tab. I highly recommend that you browse all your settings in order to get a better understanding of the actual scope of settings and customization.

Evaluating individual requirements

Evaluating your individual requirements is a crucial part of customizing your Zendesk environment. As with most systems, there are various ways to achieve a desired outcome, each with its own benefits and drawbacks. It is important to understand that the scalability of your setup highly depends on the early planning phase, allowing for a carefully structured implementation. Therefore, a great understanding of your overall requirements is paramount.

Our example company

Before we can go through the process of such an evaluation, we need to establish a far-reaching scenario by creating a fictitious example company. For the sake of simplicity and not having to refer to the company as an example throughout this book, we will call it ExampleComp.

ExampleComp is a German tech company selling their own software online. They also offer some hardware created by a third-party in collaboration with ExampleComp in order to guarantee better compatibility and support.

Next to offering their software for individual purchase, customers can choose a yearly subscription allowing them access to the full range of software solutions. These customers are called VIPs and can purchase the hardware to a discounted price. As part of the subscription, VIP customers are supposed to receive faster responses from the support team.

ExampleComp is still considered a start-up and cannot afford to provide customer-service in more than two languages (English and German), but are planning to offer support in more languages later on.

Desired setup

Note that we are not focusing on creating the best support structure possible. Our focus lies on creating a believable support structure, while making sure that we can cover the important topics later on when it comes to the actual implementation. This is also meant to be a guide for the initial planning phase. Also note that we do not plan every aspect of our Zendesk setup beforehand. Some logistics will be added later on. Just like in a real-life scenario, the requirements can change as the business grows and changes.

Keeping that in mind, let's move to the planning phase. Before planning and outlining anything specifically related to Zendesk, we should spend some time on the overall desired setup, only keeping in mind that we are using a ticket-based system.



Tip for the planning phase: As in our case, it can be helpful to plan the desired setup without keeping Zendesk's capabilities in mind. Not having any system-related restrictions to worry about can produce better ideas. While this can lead to having to adjust plans slightly later on, Zendesk offers great support for developers and lots of ideas can easily be facilitated. With too many restrictions in mind, some of the best ideas would never see the light of day.

A hypothetical customer trying to contact the support, can be a great starting point. We can work our way through the customer's experience from there. This will allow us to *draw* all the possible branches leading toward the desired outcome. We can then use the created tree to plan our Zendesk workflows.

The initial contact/channels

While customers should have different options when trying to contact support, in many cases companies decide to limit the options dramatically in order to reduce the number of support tickets.

In our case, ExampleComp wants to give their potential customers the freedom to contact support anytime during their customer journey and decides to open the following channels:

- Email
- Twitter
- Facebook
- Widget
- Help Center / Support Form

Having decided on the desired channels, we can now decide where to place those channels on our customer's journey and add some more describing details.

Email

ExampleComp distributes their software through many different online outlets, most of which do ask for a support e-mail address. Besides providing it to third-party stores, ExampleComp also plans on listing the e-mail address on their own website as well as referring to it in their printed manuals.

Twitter

ExampleComp is planning to create a dedicated Twitter account that will mainly deal with support-related issues. They want to refer to that account within the description of their company's Twitter account. This dedicated Twitter account can then also be used by the operations department in order to broadcast operational updates.

Facebook

When it comes to Facebook, it has been decided, that any direct message to ExampleComp's Facebook page will be handled as a customer support request.

Widget

A widget should also be present on their website, allowing visitors to send support requests in a few clicks.

Help Center / Support Form

ExampleComp aims to allow users to find most of the solutions within their Help Center, hoping to prevent as many tickets as possible. The Help Center will also allow users to fill out a support form in order to create a ticket.

The ticket's journey

Now that we have established the channels ExampleComp would like to use, we can start shifting our attention toward each ticket's individual journey.

First of all, we need to remember that not all channels should be treated equally. Meaning that different channels will require different SLA rules. It is a common practice, for instance, that support requests created via Facebook are being escalated quicker than tickets created via the e-mail channel. The reason for that seems to lay with the customer's expectation of service depending on the channel used.

Keeping that in mind, we can quickly sort our channels list by their supposed priority into the following groups:

1. Facebook, Twitter
2. Email, Widget, Support Form

Knowing that group 1 and group 2 have to be handled differently when it comes to escalation rules, we can take note that both groups have to be marked differently when the ticket is being created. Let's name each group. In this case, since we only have two different groups, we do not have to be too descriptive when naming them:

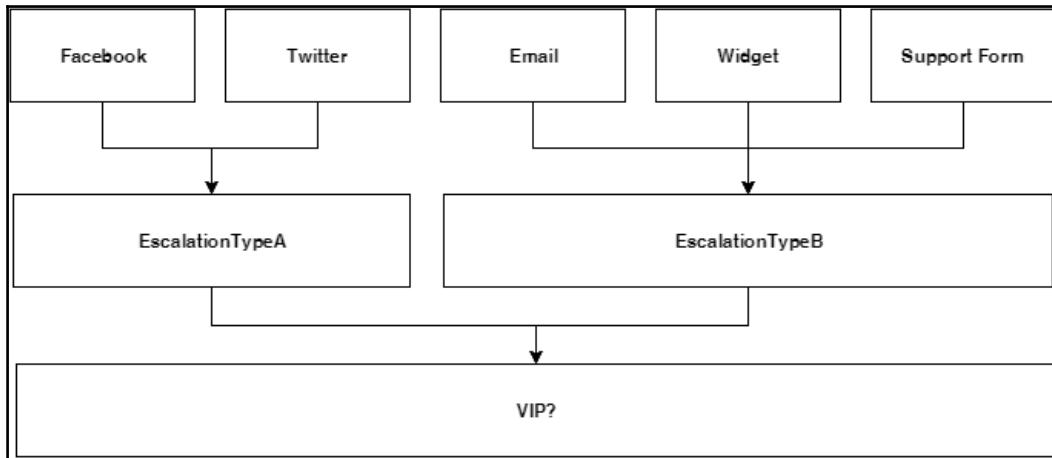
1. EscalationTypeA
2. EscalationTypeB

Next, we need to remember our VIP customers, whom ExampleComp promised special treatment when it comes to support requests. We could simply adjust our business rules to escalate those tickets even quicker than our normal tickets, by creating a third group. In this case however, we will simply funnel them into a separate view. That way we can assign special agents to only serve our VIP customers. We also allow those agents to get accustomed to our most valued customers.

But how do we tell normal customers apart from VIP customers?

At this stage, we will not worry about the actual implementation, but note that VIP users need to be marked as such. We should make a habit out of collecting these kind of questions as we will revisit them later throughout the process.

So our current ticket flow would look a little like that in the following figure:



First, each ticket is assigned an escalation type depending on the channel. After the tickets have been marked accordingly, the system should then commence marking those tickets that were created by a VIP customer.

While this seems to be a very rough concept only scratching at the surface of our final setup, this is the level of detail we want to focus on right now.

Ticket views

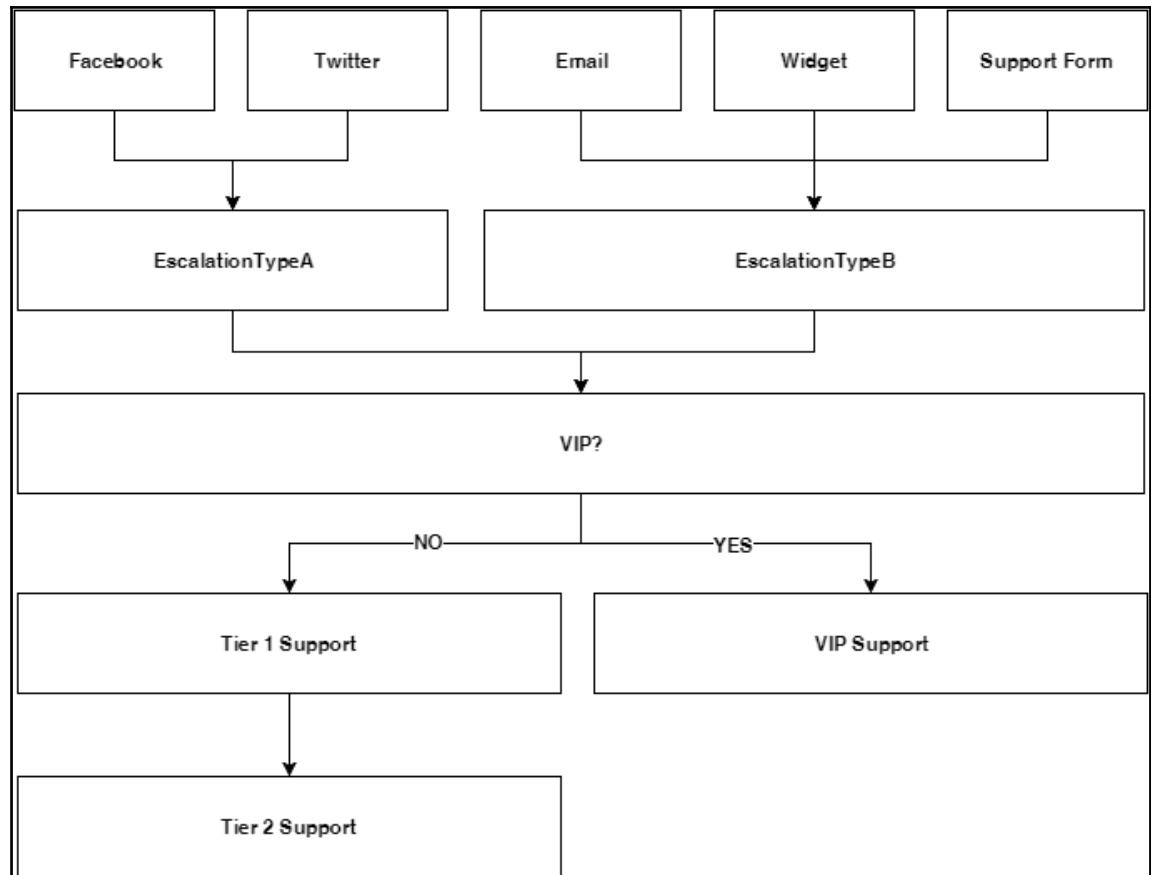
Next, we can move on to our ticket views, which we will need in order to divide and pool certain tickets together. In this case, it was decided to not separate the views by languages. While ExampleComp offers support in two languages, it currently only hires bilingual agents.

That is why we only split our support into three main categories:

- **Tier 1 Support**
- **Tier 2 Support**
- **VIP Support**

We should keep in mind that ExampleComp will offer support in more languages later on. It should be made fairly easy to divide tickets depending on the requester's language settings in the future.

Our updated ticket flow would look a little like the following figure:



While VIP tickets will land directly in the corresponding VIP Support view, all non-VIP tickets will initially land in the Tier 1 view. Agents can then decide to push them into the Tier 2 view if the customer needs more technical help.

Sorting all our tickets by priority within their corresponding views, we can leave the actual sorting process to our business rules.

Agent workflow

Having created a rough sketch of our initial ticket flow from the moment where a ticket is created up to the point where it can be picked up by agents, we can now move on to a rough outline of the agents' workflow. That also entails planning what happens to the ticket once one of the agents replied to it.

Let's start by creating a quick numbered list of what a typical workflow for a Tier 1 agent would look like:

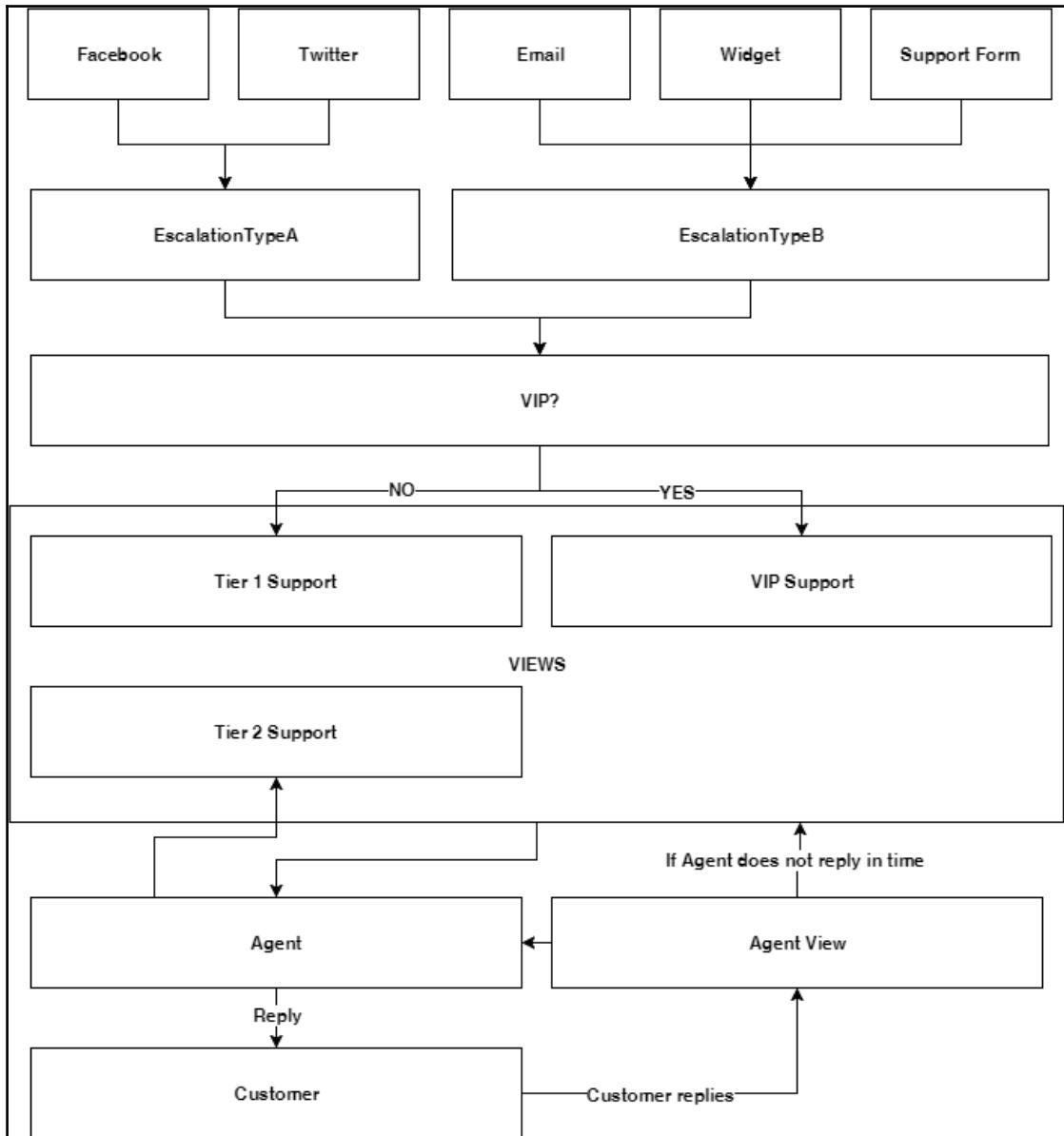
1. Opening the assigned view.
2. Picking the first ticket with the highest priority.
3. Answering the ticket or pushing it to Tier 2 support.
4. Moving on to the next ticket.

Looks easy enough. But what happens if a customer replies? There are many ways to handle customer replies. Let's review the two most common ways to do so:

- The ticket can simply be reopened and made available in its initial view, allowing any agent to pick it up in order to commence support.
- The ticket can still be assigned to the same agent, who will then commence helping that customer if possible. The agent can receive an e-mail notifying them about the customer's reply. But what if our agent stopped working already? Well, a business rule can remove the assignee (assigned agent) after a specific amount of time since the last ticket update. Doing that would force the ticket back into its initial view where another agent can then take care of the customer.

For the sake of adding a bit more complexity, we will go for the second option.

Let's take a look at our updated ticket flow, shown in the following figure:



Having added a little more complexity, we should break down the process into a few steps:

1. A customer creates a ticket via one of our open channels.
2. Depending on the channel, tickets are assigned to one of the two escalation types.
3. If our customer is a VIP, we mark the ticket accordingly.
4. The ticket is now either in the Tier 1 or our VIP view.
5. An agent opens the ticket and decides whether to answer or push the ticket to the Tier 2 view for more complex support requests.
6. The customer receives a reply.
7. If the customer replies, the ticket shows up in the agent's own view. The agent receives an e-mail notification.
8. If the agent does not update the ticket with a given time frame, the ticket is moved back to its initial view.
9. Another agent can now pick up the ticket.

Obviously, we are still looking at a very rough outline of our support's workflow. We are only describing what should happen to our tickets on the surface. However, creating such a simplified version has a lot of advantages. The ticket flow can be understood without a deeper knowledge of Zendesk. Other departments can weigh in at this point and the process can be reworked as many times as necessary until we know what the overall desired outcome should look like.

For instance, we remember that ExampleComp offers their own hardware devices. Those products are being manufactured by a third-party company. That third-party company has to use ExampleComp's software on a regular basis. This practice regularly results in support requests by that third-party company. What if, in a meeting, it is decided that those requests should not be handled the same way as other customers' requests? As we are still in the process of planning our overall workflows, we can now easily add that suggestion to our overall concept and communicate the revised plans to the management. In this case, we can simply add another step where we check if the ticket was sent by our third-party company. If so, the ticket would be marked accordingly and show up in a dedicated view. Everyone involved understands the new plan right away and we can move on to finalizing the concept.

At this point, I would like to suggest that you create a new version of the flowchart by adding the new option discussed earlier.

Zendesk setup

It is time to take our rough sketch apart and to start thinking in Zendesk terms. How does Zendesk allow us to achieve our desired setup? Looking at our flowchart, we can now take each element and review our options within Zendesk. We can therefore start to create a more detailed list of our requirements. Meaning, we can start creating a list of features that need to be set up in order to turn our plan into reality.

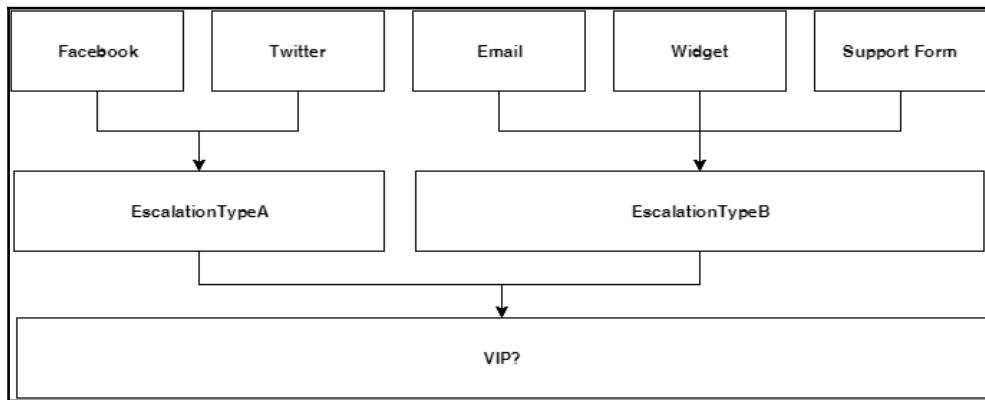
We should already keep in mind that everything we do now will help us create a quick road map later on. That is why we need to be specific enough so that a list of tasks can be generated without having to do any extra research.

It is also important to note that it might not always be as simple as working through the created flowchart from A to B. More likely, you will end up jumping around a little. This is due to the fact that each element has to fit together at the end. Planning the integration of one element might effect the way we need to set up another. Also, keep in mind that throughout this book, we are going to cover a lot of Zendesk topics in detail. A lot of these topics are mentioned before we cover them in all of their complexity. There is no need to understand everything until we get to the point of implementation later on.

Asking questions

Before we can create our final list of tasks as well as our road map, we should review our work so far. While some implementation tasks might become quite obvious, such as setting up the right views, looking at our plan we might wonder how we can achieve certain objectives. For now, we will only focus on those parts. Obviously, this is just an example. When planning your own Zendesk setup, you might already know all the answers, but the questions might be different altogether.

In order to come up with the right questions, let's start by going through our flowchart while keeping the overall process in mind:



The initial questions that come to mind are:

How can we actually know if a ticket requester is a VIP? How can we know anything about our customers when they contact us?

Clearly, it is not possible to get customer information in every scenario. In many cases, requesters might not even be customers yet. Therefore, we should focus on those cases where we can get the information and choose the right solution for us. In our case, there are two different types of information that come to mind:

- Information that will help us classify the ticket. In our case, is the user a VIP?
- Information that will help us solve the ticket, such as what software did the user buy?

While we do need to know if a user is in fact a VIP *while the ticket is being created*, it would be totally fine to receive information such as “products bought by the customer” *when first opening the ticket*. We already know, at this point, it will not be possible to get this information via a ticket created through Facebook, Twitter, or the standard Zendesk widget.

We could, however, allow the VIP information to be transferred to our Zendesk environment when a customer is using our support form. As we have the option to program on our own or change the existing support form, we can easily achieve this. Another way of achieving this would be using the **Single Sign-On (SSO)** technology for our end-users, which would allow us to set user tags. There are many ways that would lead us to the same outcome, some of which we will cover later in this book.

If we want to know more about our customers while answering their requests, it would be helpful to create our own little Zendesk app to help us out. Most companies have a unique identifier for each customer. In our case, we could use the customer's e-mail address. We could either display a generated link, that would open the companies backend, displaying the necessary information or we could go one step further and query those information from our servers and then display them within our ticket view itself. We will cover both options later in this book.

As a result of our two preceding questions, we can record the following:

Information that will help us classify the ticket. In our case, is the user a VIP?

- We can send the VIP information via Support Form
- We can use SSO and set the VIP status via a user tag
- We could set the user tag via API when the subscription is bought

Information that will help us solve the ticket, such as what software did the user buy?

- We could program an app that would generate a link to our backend
- We could program an app that would query all the information and display them directly in our app or ticket view

Our next questions would be:

How can we classify tickets as either EscalationTypeA or EscalationTypeB? How could we escalate these tickets differently depending on those two types?

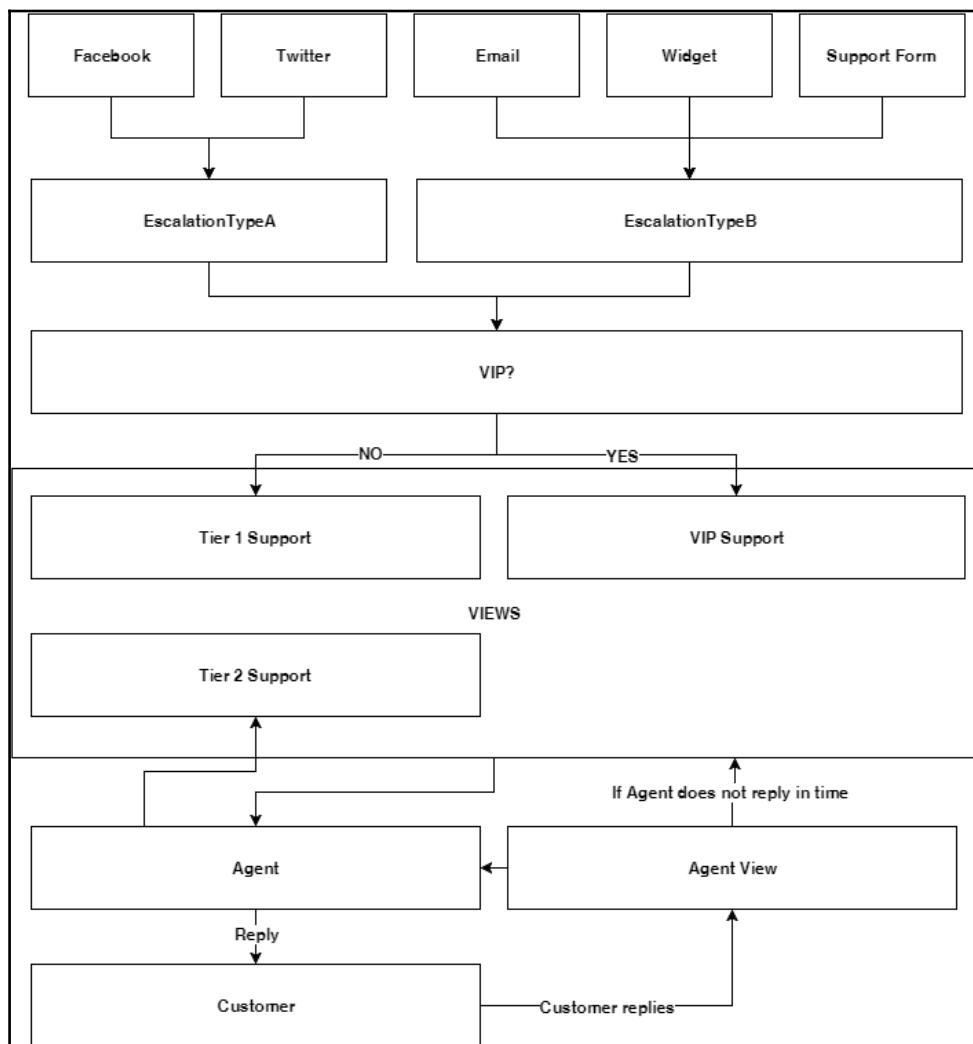
In order to do that, we could simply create two Zendesk triggers and a custom ticket field. The first trigger would check whether the ticket was created via a social media channel. If so, it would set the custom ticket field to EscalationTypeA. The second trigger would check whether the ticket was created via any of the other listed channels. In that case, it would set the custom ticket field to EscalationTypeB. Both triggers could also assign the initial priority.

In order for the tickets to be escalated depending on the type, we can make use of automations. By setting up a few automations for both cases, we can easily check the type of the ticket and escalate the ticket after a certain amount of time. Here is an example automation:

Name: Escalation Rule – EscalationTypeA – Ticket age > 1 hour -> Escalation Level: High

Knowing how we would like to integrate our business rules for the purpose of ticket escalation, we can now decide on the exact rules and create a list of triggers and automations we will need to create.

Before we do that, however, let's take a look at our complete flowchart again in order to come up with the next questions:



Looking at our flowchart, the next question that comes to mind is:

How can we allow agents to push tickets to the Tier 2 view?

Right off the bat, I can come up with three different ways to achieve that. All of which will need a ticket field that either says `tier1` or `tier2`. In any case, we will need to create a trigger that sets the ticket field to `tier1` as soon as a ticket has been created.

Now we can allow the agent to do three different things:

- The agent can set the drop-down to `tier` and submit the ticket as open.
- The agent can use a macro called `Push to Tier 2`, which we would need to create. The macro would then automatically set the right parameters.
- The agent can simply click on a button named **Push to Tier 2**. For that we would need to create a little ticket sidebar app.

While I prefer the third option because of speed, we might want to start with option 2 and integrate option 3 later on.

Listing tasks

Having found all the answers to our questions, either by online research or having used this book as a guide, we can move on and create a quick list of tasks. We can do this using our flowchart, the answers to our questions, and the overview of a basic Zendesk setup. So let's start listing our main tasks without any details and without worrying about a specific order.

We need to set up the following:

- Channels
- Custom fields
- Views
- Business rules
- Agent roles
- SLAs
- Macros
- Global Zendesk settings / Security settings
- Reporting
- Zendesk apps

Looking at each element on our list, we should already be able to envision all the subtasks and how everything snaps together as a whole. If not, it is probably best to review the flowchart and go through all the steps of the initial planning phase again as well as to review all the elements of a basic Zendesk setup.

Creating the road map

Finally, we can move on to creating our road map, which will somewhat mirror the structure of the coming chapters of this book. However, we will not only cover the specific implementation in each chapter, but also dissect each topic further and build a more complete understanding of the subject matter. That is why each chapter will cover more than the items listed in our road map. It will therefore allow you to use this book as a work of reference.

Our road map will consist of a list of tasks and subtasks sorted by the order of implementation. Our main goal should be covering all the necessary elements while making sure that we do not have to jump back and forth. All the jumping should have been done in the initial planning phase. We do, for example, set up custom fields before setting up business rules. This is due to the fact that business rules will require the existence of these fields. Let's take a quick look at the actual Zendesk environment and focus on its individual elements:



I highly suggest that you review all your elements in order to identify any dependencies. You will need to take them into account during the setup.

Let us have a look at our list of tasks:

- Setting up agent roles and groups:
 - Admin role
 - Support agent role
 - Tier 1 group
 - Tier 2 group
 - VIP group
- Creating custom fields:
 - Escalation Type
 - VIP
- Setting up channels:
 - Email

- Facebook
- Twitter
- Widget
- Support Form
- Setting up business rules / SLA:
 - Triggers
 - Automations
 - SLAs
- Creating custom apps:
 - Tier 2 App
 - Customer Information App
- Reporting via GoodData
- Security Settings and SSO:
 - General Security Settings
 - SSO

You might have noticed that I left out the exact Business rules and SLAs. At this point, I encourage you to try listing all the necessary subtasks yourself.

If you have not yet set up your Zendesk at all, you may want to apply the whole process to your own situation and come up with your very own road map.

Summary

In this chapter, we went through the very basics of a Zendesk setup and its different elements. By following a simple example, you learned how to evaluate individual requirements and how to create a road map for the final implementation.

In the next chapter, you will learn everything there is to know about agent roles, groups, organizations, and user tags. We will also cover different methods to import existing user databases. After covering these topics, we will revisit our example company and apply some of the knowledge you have learned.

2

Agent Roles, Groups, Organizations, and User Tags

When it comes to working with an environment such as Zendesk, which was built to communicate with millions of customers, it is absolutely crucial to understand how we can manage our user accounts and their tickets without losing track of our processes. However, even when working with a smaller customer base, keeping scalability in mind, we should apply the same diligence when it comes to planning our agent roles, groups, organizations, and user tags.

Not only do we allow more efficient workflows to be planned and added in the future, but a carefully planned setup of your agent roles, groups, organizations, and user tags will also enable us to create more detailed and meaningful performance reports.

That is why, in this chapter, we will examine each of the options provided by Zendesk and dissect their individual capabilities. We will review where these options fit in our overall setup and how we can implement our desired workflows.

At the end of this chapter, you will know what part agent roles, groups, organizations, and user tags play in our overall setup and how to utilize them to achieve our goals.

This chapter will cover the following topics:

- Users/agents/custom agent roles
- Groups
- Organizations
- User tags
- Importing existing user databases (CSV file, Zendesk API)

Users/agents

In Zendesk, agents are just like end users and are classified as users. Both can be located in the same pool of listed accounts. The difference, however, can be found in the assigned role. The role defines what a user can or cannot do. The end users, for example, do not possess the necessary rights to log in to the actual helpdesk environment. Easily enough, the role for end users is called **end-user**.



In Zendesk, **users** are also referred to as **people**. Both are equivalent terms. The same applies to the two terms **end-users** and **customers**.

You can easily access the whole list of users by following these two steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **People** located under **MANAGE** within the admin menu:

The screenshot shows the Zendesk Admin interface. On the left, there is a blue sidebar with various icons and a list of management options. A red box highlights the 'Manage' section, and a red number '2' is placed next to the 'People' link under it. Another red box highlights the gear icon in the sidebar. The main content area is titled 'People' and contains a brief description of what users are. It features a search bar with a placeholder 'Enter the first few letters of the search term' and a 'Search' button. Below the search bar is a link to 'add user | group | organization | role'. A 'Bulk Management' sidebar on the right provides links for importing users and organizations via API.

Unlike for end-users, there are a few different roles that can be assigned to an agent. Out of the box, Zendesk offers the following options:

- Agent/Staff
- Team leader
- Advisor
- Administrator

While the **agent** and **staff** roles come with the necessary permissions in order to solve tickets, the **team leader** role allows more access to the Zendesk environment.

The **advisor** role, in contrast, cannot solve any tickets. This role is supposed to enable the user to manage Zendesk's workflows. This entails the ability to create and edit automations, triggers, macros, views, and SLAs.

The **admin** role includes some additional permissions, allowing the user to customize and manage the Zendesk environment.



Note: The number of available roles depends on your Zendesk plan. The ability to create custom roles requires the Enterprise version of Zendesk. If you do not have the option to create custom roles and do not wish to upgrade to the Enterprise plan, you may still want to read on. Other plans still allow you to edit the existing roles.

Custom agent roles

Obviously, we are scratching on the surface here. So let's take a closer look into the roles by creating our own custom agent role.

In order to create your own custom role, simply follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **People** located under **MANAGE** within the admin menu.
3. Click on **role** located at the top of the main area (next to **add**):

The screenshot shows the Zendesk Admin Home interface. On the left, there is a sidebar with various management options like Overview, Apps, Marketplace, Manage, Channels, and a Feedback Tab. A red box labeled '1' highlights the Admin icon (gear symbol). Another red box labeled '2' highlights the 'People' link under the 'Manage' section. At the top right, there is a red box labeled '3' highlighting the 'role' link next to the 'add user | group | organization' buttons. The main content area is titled 'People' and contains a search bar and a list of users with edit and delete icons.

The process of creating a custom role consists of naming and describing the role followed by defining the permissions:

A screenshot of the 'Role creation' form. It has two main sections: 'Role name' containing a text input field with the value 'A Custom Role Example', and 'Description' containing a text area with the placeholder text 'This role serves as an example to showcase the process.'

Permissions are categorized under the following headings:

- Tickets
- People
- Help Center
- Tools
- Channels
- System

Each category houses options to set individual permissions concerning that one specific topic. Let's examine these categories one by one and decide on each setting for our example role - **Tier 1** agent.

Ticket permissions

In the first part, we can choose what permissions the agent should receive when it comes to handling tickets:

What kind of tickets can this agent access?

- Those assigned to the agent only
- Those requested by users in this agent's organization
- All those within this agent's group(s)
- All

Agent can assign the ticket to any group?

- Yes
- No

What type of comments can this agent make?

- Private only
- Public and private

Can edit ticket properties?

- Yes
- No

Can delete tickets?

- Yes
- No

Can merge tickets?

- Yes
- No

Can edit ticket tags?

- Yes
- No

Tickets	<p>What kind of tickets can this agent access?</p> <p>All</p> <p>What type of comments can this agent make?</p> <p>Public and private</p> <p><input checked="" type="checkbox"/> Can edit ticket properties Enables the agent to edit properties for any accessible ticket. Without this permission, the agent cannot be assigned to a ticket.</p> <hr/> <p><input type="checkbox"/> Can delete tickets Enables the agent to delete any accessible tickets. Also, it permits the agent to mark a ticket as spam if they also have permission to delete the end user.</p> <p><input type="checkbox"/> Can merge tickets Allows the agent to merge any two accessible tickets.</p> <p><input type="checkbox"/> Can edit ticket tags Provides the agent with the ability to add and remove tags on tickets. Agents without this ability can still set custom fields, but cannot add tags with macros.</p>
---------	--

So what kind of tickets should this agent be able to access?

Our Tier 1 agent should only work on tickets within the Tier 1 group. Therefore, we will go for the option “All those within this agent's group”.

Should the agent have the ability to assign tickets to other groups?

According to our flowchart from *Chapter 1, Configuring Your Own Zendesk*, we want the Tier 1 agents to push tickets to the Tier 2 view if necessary. As this would entail changing the ticket's group, we should check this option.

What type of comments should we allow this agent to make?

As comments constitute the actual reply to our end-users and our Tier 1 agents are indeed supposed to answer requests, we will need to go for **Public and private**. This will allow our Tier 1 agents to reply to our end-users.

Should the agent be allowed to edit ticket properties?

This permission is necessary if we want our agent to be assigned tickets. As we want our agents to reply to customers and consequently take ownership of the ticket, we will check this option.

Should our agents be allowed to delete tickets?

As a general rule, tickets should never be deleted permanently. Zendesk cannot recover deleted tickets. In order to avoid any issues, we will choose to not check this box.

Should our agents be allowed to merge tickets?

In many cases, end-users send more than one ticket concerning the same issue. If that happens, merging tickets can be helpful. We will set this permission and check the box.

Should our agent be able edit ticket tags?

Ticket tags can be very helpful. We might want to create macros that automatically add tags to a ticket upon solving. That is why we will set this permission as well.

People permissions

The second part allows us to set permissions regarding the agent's ability to manage other users/people:

What access does this agent have to end-user profiles?

- Read only
- Add and edit within their organization
- Add, edit, and delete all

May this user view lists of user profiles?

- Cannot browse or search for users
- Can view all users in your account

Can add or modify groups and organizations?

- Yes
- No

People

What access does this agent have to end-user profiles?

Add, edit, and delete all ▾

End-user editing rights enable the agent to verify and assume end-users. Note that only admins can change a user's role.

May this user view lists of user profiles?

Can view all users in your account ▾

This setting does not affect the agent's ability to add end-users or edit their profiles.

Can add or modify groups & organizations

Groups categorize agents while organizations enable you to segment your end-users.

So what kind of access should our agent have to end-user profiles?

For now, we will go for option one and choose **Read only**. It would make sense to forward more complicated tickets to our Tier 2 support, who receive the permission to edit end-user profiles.

Should our agent be allowed to view the full list of users?

In some cases, it might be helpful if agents can search for users within the Zendesk system. In this case, we will answer our question with a yes and check the box.

Should the agent be allowed to modify groups and organizations?

None of our planned workflows seem to require this permission. We will not check this box, and therefore remove another possible source of error.

Help Center permissions

The third part concerns the **Help Center** permissions:

Can manage Help Center?

- Yes
- No

Help center

Can manage Help Center

Does our agent need the ability to edit the Help Center?

As the primary task of our Tier 1 agents consists of answering tickets, we will not check this box and leave this permission to our administrators.

Tools permissions

The fourth part gives us the option to set permissions that allow agents to make use of Zendesk Tools:

What can this agent do with reports?

- Cannot view
- Can view only
- Can view, add, and edit

What can this agent do with views?

- Play views only
- See views only
- Add and edit personal views
- Add and edit personal and group views
- Add and edit personal, group, and global views

What can this agent do with macros?

- Cannot add or edit
- Can add and edit personal macros
- Can add and edit personal and group macros
- Can add and edit personal, group, and global macros

Can access dynamic content?

- Yes
- No

<p>Tools</p> <p>What can this agent do with reports?</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">Can view, add and edit ▾</div> <p>Data export capabilities is included with the ability to edit reports.</p> <p>What can this agent do with views?</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">Add and edit personal, group, and global views ▾</div> <p>Views define preset conditions for a collection of tickets. Views can be personal, for group(s), or for all agents in your Zendesk.</p> <p>What can this agent do with macros?</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">Can add and edit personal, group, and global macros ▾</div> <p>Macros apply pre-defined actions to a ticket. Macros can be personal, for group(s), or for all agents in your Zendesk.</p> <p><input type="checkbox"/> Can access dynamic content</p> <p>Can review, add, and edit dynamic content.</p>

Should our agent have the permission to view, edit, and add reports?

We do not want our agents to interact with Zendesk's reports on any level. We might, instead, want to create custom reports via GoodData, which can be sent out via e-mail to our agents. Therefore, in this case, we choose the **Cannot view** option.

What should the agent be allowed to do with views?

As we will set up all the necessary views for our agents, we will go for the **See views only** option. If there is a need for private views later on, we can always come back and change this setting retroactively.

What should the Tier 1 agent be allowed to do when it comes to macros?

In our example, we want to create a very streamlined support. All creation of content should take place at the administrative level and be handled by team leaders. Therefore, we will select the **Cannot add or edit** option.

Should the agent be allowed to access dynamic content?

We will not check this option. The same reasons apply here: content creation will happen at the administrative level.

Channels permissions

The fifth part allows us to set any permissions related to ticket channels:

Can manage Facebook pages?

- Yes
- No

Channels	<input type="checkbox"/> Can manage Facebook Pages Can add Facebook Pages that pull Wall posts as tickets.
----------	---

There is no need for our Tier 1 agents to receive any of these permissions as they are of an administrative nature.

System permissions

Last but not least, we can decide on some more global system-related permissions:

Can manage business rules?

- Yes
- No

Can manage channels and extensions?

- Yes
- No

System	<input type="checkbox"/> Can manage business rules Able to manage triggers, automations, and SLA targets.
	<input type="checkbox"/> Can manage channels and extensions Channels are modes of communication such as chat, email, and Twitter. Extensions include widgets, targets, and integrations.

Again, there is no need for our Tier 1 agent to receive these permissions as they are of an administrative nature.

Custom roles for our example

At this point, you might want to create all the necessary agent roles. There is no need to worry about stuffing up. All roles can easily be altered again later on. Here is a list of agent roles we might need:

- Tier 1
- Tier 2
- VIP
- Administrator

It may seem unnecessary to create all these roles. The Tier 2 and VIP role might even share the exact same permissions. However, in the future, it could be helpful having the option to add specific permissions for one particular group of agents without having to create a new role and reassign agents.

Groups

Groups, unlike organizations, are only meant for agents and each agent must be at least in one group. Groups play a major role when it comes to support workflows and they can be used in many different ways. How to use groups becomes apparent when planning your support workflow.

In our case, we have four types of support tickets:

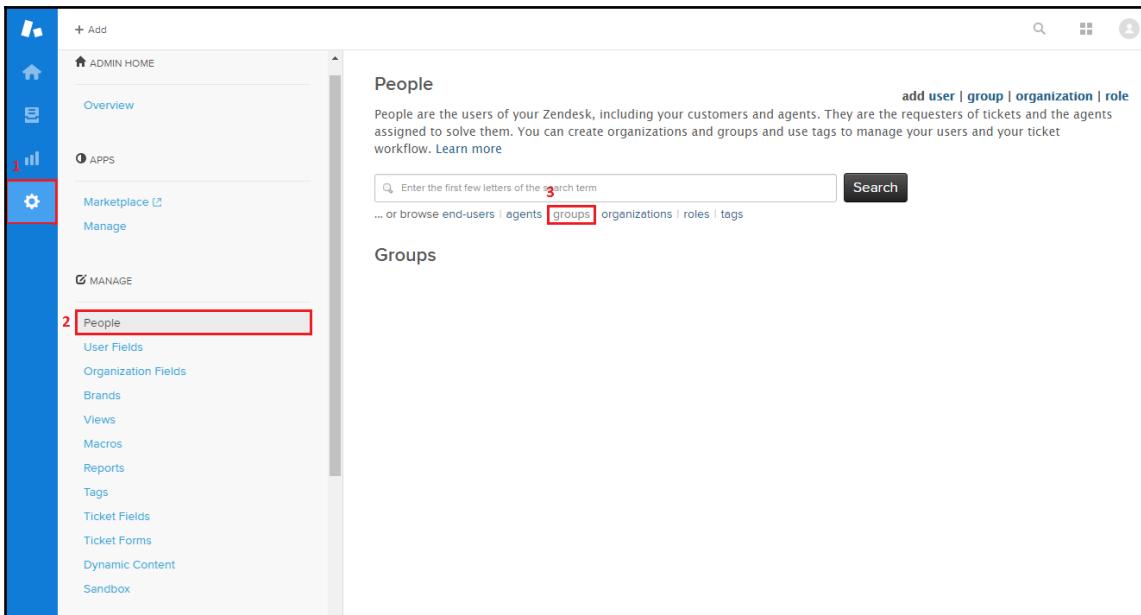
- Tier 1 Support
- Tier 2 Support
- VIP Support
- Internal Support

Each type of ticket is supposed to be answered by specific agents only. In order to achieve this, we can create one group for each type of ticket and later assign these groups to our agents accordingly.

In order to review and edit already existing groups, simply follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **People** located under **MANAGE** within the admin menu.

3. Click on **groups** located under the search bar within the main area:



As per our example, we will need four groups. In order to add a new group, simply follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **People** located under **MANAGE** within the admin menu.
3. Click on **group** located at the top of the main area (next to **add**):

The screenshot shows the Zendesk Admin interface. On the left, there is a sidebar with various sections: ADMIN HOME (Overview), APPS (Marketplace, Manage), MANAGE (highlighted with a red box, containing People, User Fields, Organization Fields, Brands, Views, Macros, Reports, Tags, Ticket Fields, Ticket Forms, Dynamic Content, Sandbox), CHANNELS (Email, Twitter, Chat, Facebook, Voice, Feedback Tab). At the bottom of the sidebar is a gear icon. The main content area is titled 'People' and contains a search bar and navigation links for end-users, agents, groups, organizations, roles, and tags. Below this is a section titled 'Groups' with a table header and several rows of data, each with an 'edit' link. The 'group' button at the top right of the main area is also highlighted with a red box and has a small number '3' above it.

Creating a group is easy. We simply choose a name and tick the box next to each agent that we would like to be associated with this group:

Group name

Enter a name for this group, then select the agents to include.

Groups are collections of agents set up to support your ticket workflows. [Learn more](#)

Agents in group

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

[Create group](#)



There are two ways to add an agent to a group. While you may choose to navigate to the group itself in order to edit it, you can also assign groups to agents within their own user panel.

Organizations

Organizations can be very helpful when managing workflows, though there is no imperative need to associate end-users with an organization. Therefore, we should ask ourselves this: do we need to use organizations to achieve our desired workflows?

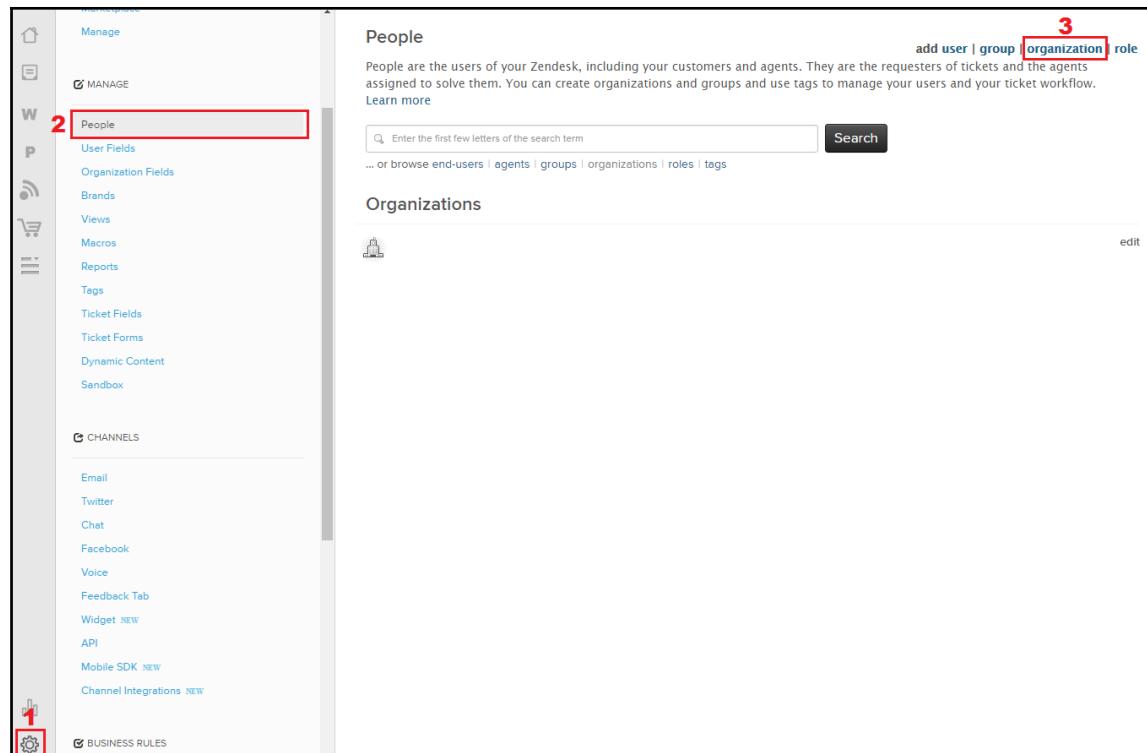
Before we can answer this question, let's have a look at how organizations work in Zendesk.

When creating an organization within Zendesk, you may choose one or more domains associated with that organization. As soon as an end-user creates a ticket using an e-mail address with that specific domain, the user is added to that organization.

There are a few more things you can set within an organization. So let's have a quick look at all the available options.

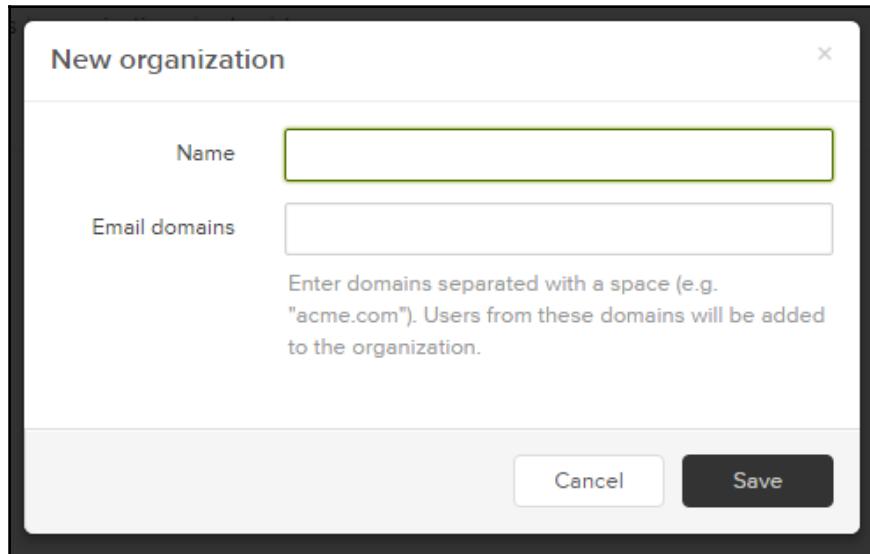
In order to add a new organization, simply follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **People** located under **MANAGE** within the admin menu.
3. Click on **organization** located at the top of the main area (next to **add**):



When adding a new organization, Zendesk asks you to provide the following details:

- The name of the organization
- The associated domains



Once we click on **Save**, Zendesk automatically opens this organization as a tab and shows next to any ticket associated with the organization. Here are a few more options we can set up:

- Tags
- Domains
- Groups

- Users
- Details
- Notes

Tags	-
Domains	test.com
Group	-
Users	Can view own tickets only
Details	-
Notes	-

Tags

Zendesk allows us to define tags, which would automatically be added to each ticket, created by a user within this organization.

Domains

We can add as many associated domains as we need. Each domain should be separated by a single space.

Group

Tickets associated with this organization can be assigned to a group automatically. We can choose any group via a drop-down menu.

Users

We get the following two options to choose from:

- **Can view own tickets only**
- **Can view all org tickets**

This allows us to enable users, who are part of this organization, to only view their own tickets or to review all the tickets created by users within this organization. If we choose for users to view all the tickets within their organization, we receive two more options:

- **...but not add comments**
- **...and add comments**

Details

We may add additional information about the organization such as an address.

Notes

Additionally, we may add notes only visible for agents.

Organizations for our example

Now that we know how organizations work in Zendesk, let's have another look at our initial question: do we need to use organizations to achieve our desired workflows?

The answer is yes. When looking back at our desired workflow, we remember the third-party company that manufactures hardware for ExampleComp. During the initial planning phase, we realized that this third-party company would need the help from our support team as well. This is the solution that we came up with in [Chapter 1, Configuring Your Own Zendesk](#):

In this case, we could simply add another step where we check if the ticket was sent by our third-party company. If so, the ticket would be marked accordingly and show up in a dedicated view.

Now that we have the necessary tools in order to achieve our goal, we can commence setting up our organization accordingly.

You may now proceed to create a new organization (or edit the existing organization) so that tickets created by users from our third-party company end up in the dedicated group (Internal Support).

User tags

To understand user tags, we need to understand how Zendesk utilizes tags and how they can help us.

Tags can be added for users, organizations, and tickets, while user tags and organization tags will be ultimately applied to tickets when they are created.

For instance, if a user is tagged with the `vip` tag, all their tickets will subsequently be tagged with the `vip` tag as well.



We can then use that tag as a condition in our business rules.

But how can we set user tags without having to do some manually?

This is a very important question. In our flowchart, we require the knowledge whether a customer is in fact a VIP user in order for our business rules to escalate the tickets according to our SLA rules. Let's have a quick look at our plan from *Chapter 1, Configuring Your Own Zendesk*:

- We could send VIP information via Support Form
- We could use SSO and set the VIP status via a user tag
- We could set the user tag via API when the subscription is bought

In our first option, we would try to send a tag from our support form to Zendesk so that the ticket is tagged accordingly.

In our second option, we would set the user tag and subsequently the ticket tag via SSO.

In our last option, we would set the user tag via the Zendesk API when a subscription is bought. We remember that a customer of our ExampleComp becomes eligible for VIP service only on having bought a software subscription.

In our case, we might go for the third option. It is a very clean solution and also allows us to remove the user tag when the subscription is cancelled.

So how can we achieve this?

Luckily, Zendesk offers a greatly documented and easy to understand API. We can therefore do the necessary research and forward our requirements to our developers. Before we look at any code, we should create a quick outline:

1. User registers on ExampleComp's website: a Zendesk user is created.
2. User subscribes to software package: the user tag is added to the existing Zendesk user.
3. User unsubscribes from software package: the user tag is removed from the existing Zendesk user.
4. User deletes account from ExampleComp's website: the Zendesk user is removed.

All this can easily be achieved with a few lines of code. You may want to refer your developers to the following

webpage: https://developer.zendesk.com/rest_api/docs/core/users

If you have coding experience, here are the necessary code snippets:

For creating a new end-user:

```
curl -v -u {email_address}:{password}
https://{{subdomain}}.zendesk.com/api/v2/users.json \
-H "Content-Type: application/json"
-X POST
-d '{"user": {"name": "FirstName LastName", "email": "user@example.org"}}'
```

For updating an existing user:

```
curl -v -u {email_address}:{password}
https://{{subdomain}}.zendesk.com/api/v2/users/{id}.json \
-H "Content-Type: application/json"
-X PUT
-d '{"user": {"name": "Roger Wilco II"}}'
```

Importing existing user databases

In many cases, companies already have a huge amount of customers before the decision to use Zendesk has been made. In such a situation, it might be handy to have the option to import those users in bulk, especially if you like to set a user tag such as our `vip` tag.

Zendesk offers us two different options when it comes to importing users in bulk:

- CSV file
- Zendesk API

We will focus on option one as we have already covered the option to add Zendesk users, utilizing the Zendesk API when discussing user tags.

In order to locate the option allowing you to import users in bulk, follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **People** located under **MANAGE** within the admin menu.

3. Click on **Bulk user import** located on the right side of the main area (located in a grey box):



Before uploading our CSV file, we can choose one or both of the following options:

- **Create new users**
- **Update existing users**

Bulk user import

Create new users
Users that exist in the import data, but not in Zendesk, will always be created.

Update existing users
Update existing users with the attributes specified in the import data.

Send email notification to users if their password is changed?
An email notification will be sent to end-users, admins, and agents when their standard Zendesk password changes. Zendesk recommends that you enable this feature.

Select CSV file

No file chosen
Let me paste in data instead. Read about CSV format requirements on the right.

WARNING: An email will be sent to all users who are created unless you have disabled the welcome message on the End-users (customers) settings page.

Cancel or **Import**

As we are importing new users only, we can uncheck **Update existing users**.

In order for the bulk-import to work, we will need to prepare our CSV file accordingly. Zendesk states the following:

The data must be in the comma separated values (CSV) format and saved as UTF-8.

A good way to prepare such a file is using a spreadsheet program such as Microsoft Excel. Having our example in mind, we will create an example, importing the following details:

- Name
- Email
- Tags

	A	B	C
1	name	email	tags
2	test user	testuser@test.com	vip
3			

The first row, also called the header row, must follow a specific order. While we are only using three fields, you can find a full list of available fields and the accepted order at the following link:

<https://support.zendesk.com/hc/en-us/articles/203661996-Bulk-importing-users-and-organizations>

Once we are happy with our spreadsheet, we need to save it in the **Comma Separated Values (CSV)** file format on our local hard drive.

Now simply click on **Choose File**, locate the file on your hard drive, and confirm by clicking on **Import**.

After a short while, Zendesk will send you an e-mail with the subject line as “Your user import is ready”.

A quick user search in Zendesk confirms that our import was successful:

The screenshot shows the Zendesk user profile for 'test user'. The profile page has a sidebar on the left with basic information: Role (End-user), Access (Tickets requested by user), Primary email (testuser@test.com), Tags (vip), Org. (test), Language (German - Deutsch), Time zone (GMT+02:00) Vienna, Details, and Notes. Below the sidebar, it shows the creation and update times. The main content area shows the user's ticket history, which is currently empty. A red box highlights the 'test user' name in the top navigation bar.

Congratulations, you have successfully imported a user into your Zendesk environment!

Summary

In this chapter, you learned about Zendesk users, roles, groups, organizations, and user tags. Following up on our road map, we laid important groundwork for a functioning Zendesk environment by setting up some of the basic requirements for more complex workflows.

In the next chapter, you will learn about different custom fields, how they can be used, and how we will utilize them in our example scenario.

3

Creating Custom Fields

Being able to provide great customer support depends on a lot of different factors. One of them, without question, is having the necessary information needed to give the best possible answer. Therefore, it is a no-brainer that our end users should always receive the adequate tools, allowing them to supply us with these bits of information.

Equally important is that all these bits of information should be displayed in such a manner that our agents can encompass the exact nature of the request, as well as the surrounding circumstances. At the same time, we want our business rules to access these bits of information in order to make use of them.

There are user fields, ticket fields, and organization fields. In this chapter, we will cover all of them and analyze the different types of fields available for us and how we can use them in our project.

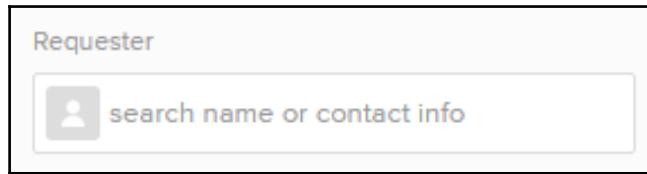
This chapter will cover the following topics:

- What are fields in Zendesk?
- Standard ticket fields (system fields)
- Different types of field
- Creating custom user, ticket, and organization fields

What are fields in Zendesk?

Fields are, simply put, containers for information. You could say that users, tickets, and organizations are objects in Zendesk; each containing a range of fields to hold the information describing the object.

Let's have a quick look at an example field:



This field called **Requester** is a ticket field as it belongs to the ticket object. Every ticket requires a requester. In most cases, the requester would be an end user.

Another great example would be the **Description** field, which holds the initial text provided by the requester of a ticket. The content of some ticket fields is supplied by the end-user who created the ticket. In direct contrast, the **Assignee** field, holding the information about what group or agent the ticket is assigned to, is set within Zendesk. In this case, the end user does not supply the information:



Standard ticket fields

Now that we have a general understanding of fields, let's have a quick look at Zendesk's standard fields for tickets.

The standard fields, also called system fields, are not custom fields, but fixed pre-existing fields that cover the most commonly required information for a helpdesk environment. While some of them are optional and can be deactivated, some are required when using Zendesk.



Note: System fields are deeply integrated into the architecture of Zendesk and should not be used for purposes other than intended. As for any environment that is constantly reworked and refined, as a general rule, things can change. For custom workflows it always pays off to create your own fields.

Requester

As previously mentioned, all tickets need a **requester**. The requester is usually an end user who created the support ticket.

Assignee

The **assignee** is usually the agent who is attending to the ticket but can also be a group. For instance, a group can be set as the assignee via a business rule to only allow agents within this group to attend to the ticket.

CCs

CCs work just the way they do when sending e-mails. This field can be deactivated within the Zendesk settings.

Share

This field is only visible when the option **Ticket Sharing** has been activated within the settings. Tickets can be shared between Zendesk setups.

Subject

The **Subject** field is usually included in the request sent by an end user. If a customer chooses to contact the support via an e-mail channel, the subject field is automatically filled with the e-mail's subject text.

Description

The **Description** field contains the initial text sent by the requester. The description is displayed as the first comment in a ticket. When using the e-mail channel, the e-mail body is used to set the description text.

Status

The status can be set to one of the following:

- New
- Open
- Pending
- On-hold
- Solved
- Closed

It describes the ticket's status within the system.

Type

The type can be set to one of the following:

- Question
- Incident
- Problem
- Task

It describes the nature of the ticket and is very useful when it comes to planning your workflows and business rules.

Priority

Priority can be set to one of the following:

- Low
- Normal
- High
- Urgent

It describes the level of urgency and can be tied to a time frame in which the ticket should be replied.

Tags

Tags are used to add additional information that can be used within your business rules and metrics. Tags play an important role when it comes to customizing Zendesk and we will be using them a fair bit later.

Standard user and organization fields

While we could commence by listing all the user and organization fields here, let's glance back at a previous chapter instead. Why? Without having known them at that time, we covered fields before. We should take this chance to put things into context.

In [Chapter 2, Agent Roles, Groups, Organizations, and User Tags](#), we covered organizations and how to set them up manually. As an organization is nothing more than an object with certain fields, we did actually cover all of them in the process:

- Tags
- Domains
- Groups
- Details
- Notes

The same goes for user fields. We covered most of them in [chapter 2, Agent Roles, Groups, Organizations, and User Tags](#), while talking about users, user tags, groups, and custom roles. Just to refresh our memory, let's list them here and put them into context as well:

- Role
- Tags
- Organization
- Details
- Notes

Some other user fields that we did not mention are self-explanatory:

- Primary e-mail
- Language
- Time Zone

A few user fields are only available when the user's role is not set to End-user:

- Groups
- Alias
- Signature

Different types of fields

Having had a quick look at all the different pre-existing ticket fields in Zendesk, we quickly notice one thing: not all of the fields are simple text fields, that can hold just any arbitrary string. There is a whole range of different field types that can be utilized in Zendesk.

Let's list the available field types before going into more detail:

- Drop-down list
- Text
- Multi-line text
- Numeric
- Decimal
- Checkbox
- Regular expression
- Date

Each type of field serves a different purpose. While some can only display a string, others can also be used for business rules and metrics.

But why is that? When can we use a field in our business rules? What is the decisive difference between the fields?

In order to understand, let's divide all field types into two groups:

- Those with predefined selectable options and
- Those that can hold an arbitrary string (even if restricted)

To keep it simple, only the following two field types have very specific predefined options to select:

- Drop-down list
- Checkbox

All the other options allow a completely free choice within the limitations of the type of field. Having predefined selectable options, however, allows us to define a tag for each possible selection. If one of the options is selected, the tag will be added to the ticket and enables us to check for it within our business rules or measure its occurrence within our metrics.

We can conclude that the Drop-down list and Checkbox field types can be used in business rules.

Let's examine each option in detail in order to understand how they could be used later.

Drop-down list

You already learned that the **Drop-down list** allows us to predefine answers, which we can then use as a condition in business rules. We can also decide for the **Drop-down list** to be displayed in the support form, allowing the end-user to select one of the options. There are a few different use-case scenarios. Overall, the **Drop-down list** is one of the most important and useful fields in our arsenal:

Drop-down list

Provide a drop-down list with options you define. The ticket will be tagged accordingly.

Favorite animal

Frog ▾

Text

The **Text** field allows for a simple single line of text. This can be useful in a lot of different scenarios. You may want to allow an end-user to give a specific answer while limiting the answer to be a short and precise string:

Text

Capture small text.

Department

Sales

Multi-line text

The **Multi-line text** field allows for multiple lines of text. You may, for instance, want to use it to store a user's mailing address:



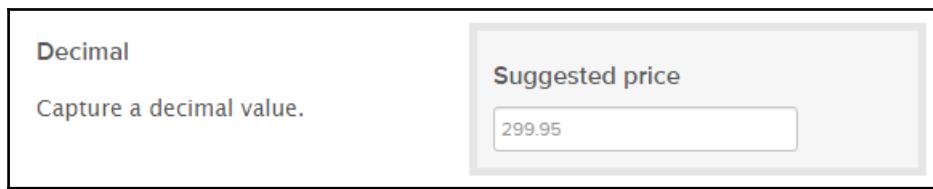
Numeric

The **Numeric** field type allows us to limit the input to integers, reducing possible mistakes:



Decimal

The **Decimal** field type allows us to give the option for decimal values:



Checkbox

The **Checkbox** type is one of the two previously mentioned options with predefined answers. In the case of **Checkbox**, the answer can only be yes or no. We may define a tag for when the box has been checked:

A screenshot of a form interface showing a checkbox field. On the left, a label "Checkbox" is above the placeholder text "Capture a yes/no value.". To the right is a larger input box with a title "May we contact you?" and a single checkbox labeled with a small square.

Date

Also very helpful is the **Date** field type. Use cases can vary from a user's date of birth to a deadline for a project:

A screenshot of a form interface showing a date field. On the left, a label "Date" is above the placeholder text "Capture a date in the future or past.". To the right is a larger input box with a title "Select a date" and a date picker icon.

Regular Expression

The **Regular Expression** field allows you to create a mask by entering a Ruby regular expression. This allows us to create a field limited to strings formatted in a particular way. This could be a registration key, product ID, or anything that follows strict syntax rules:

A screenshot of a form interface showing a regular expression field. On the left, a label "Regular Expression" is above the placeholder text "Capture input that verifies according to a regular expression that you define.". To the right is a larger input box with a title "Product ID" containing the value "A12R-OFWGKTA-3X".

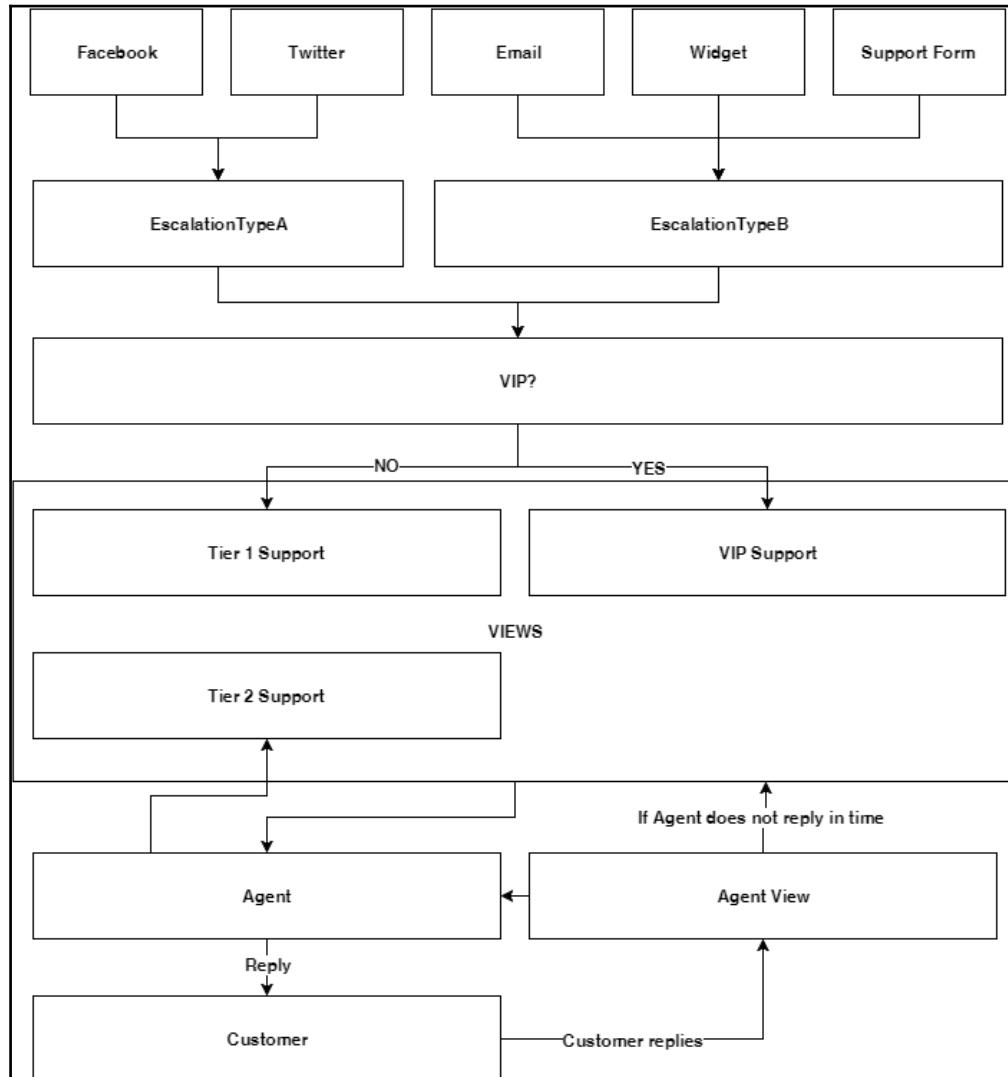


Search the Internet for “Ruby-based regular expression” and learn more about the syntax used. This can become extremely helpful and significantly reduce user errors.

Creating custom fields

So now that we have a better understanding of fields in general and what types of fields there are, we can move to creating our own custom fields.

Before we do so, however, let's decide what custom fields we will need by reviewing our flowchart and road map:



Looking at our flowchart, we can see a few possible options when it comes to custom fields. We already know that we can create custom fields for users, organizations, and tickets. So let's divide our train of thought for now.

Custom user fields

In Chapter 2, *Agent Roles, Groups, Organizations, and User Tags*, we decided to add the VIP user tag via the Zendesk API whenever a customer purchases a subscription. This still applies. But in order to visualize whether the user is a VIP, and to give our agents the option to upgrade a user to the VIP status manually without having to enter the tag, we should add a checkbox field. Setting the tag via API will still work fine and the checkbox will automatically be checked once the tag has been added. At the same time, the tag can be removed by unchecking the checkbox.

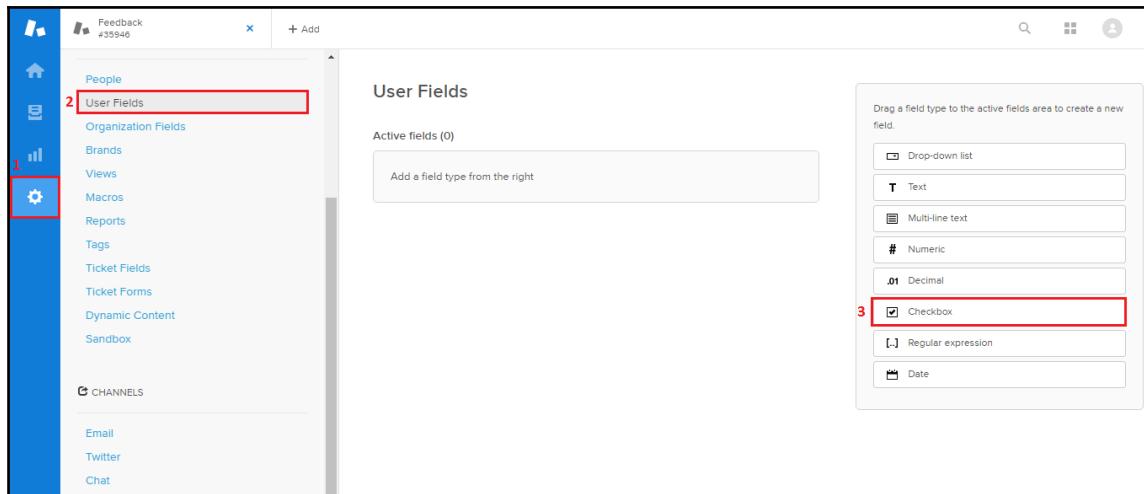


While it is true, that we can set the tag via API and that the system will subsequently check the box, it is not considered best practice. Using the tag to check the box would mean that the system would not work once we change the tag, which might become necessary due to a workflow change later. To be safe, we should use the API to check that specific checkbox using the custom field ID. In our business rules, we should also stop using conditions using the tag and check whether that specific checkbox is checked instead.

So let's go ahead and add the first custom user field:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **User Fields** located under **MANAGE** within the admin menu.

3. Click on **Checkbox** located on the right within the main area:



We are now presented with a few options that we should take care of, before clicking on **Create field**:

- Field title shown to agents
- Field key
- Description (optional)
- Tag

For **field title**, we should go for something like **VIP**. The title is self-explanatory and our agents will understand it. **Field key** serves as an identifier for when we are using placeholders or the Zendesk API. **Description** is optional, though I would suggest making use of it. It cannot hurt to add additional information. **Tag** should be the same as defined in Chapter 2, *Agent Roles, Groups, Organizations, and User Tags*, which was **vip**.

The screenshot shows a modal dialog for creating a new custom field. The field type is selected as 'Checkbox'. The 'Field title shown to agents' is set to 'VIP'. The 'Field key' is 'vip_status'. A note explains that the key is unique and cannot be changed. The 'Description (optional)' field is empty. A 'Tag' is added as 'vip'. A note says a tag can be used in views, triggers, and automations. At the bottom, there are 'Cancel' and 'Create field' buttons.

Checkbox

Field title shown to agents

VIP

Field key

vip_status

The key identifies this field in placeholders and the API. After a field is created, you cannot change its key.

Description (optional)

Tag

vip

Enter a tag to add to the organization when the checkbox is selected. You can use the tag in views, triggers, and automations.

Cancel Create field

Once we have filled the form to our liking, we can create the field by clicking on **Create field**.

A little checkbox with the title **VIP** will now show up in every user's panel. We remember that checking the box will add the little string `vip` as a user tag. If that user decides to create a ticket, the ticket will also be tagged with `vip`:



Custom ticket fields

Looking at our flowchart, we may be tempted to create a custom field for the escalation type of the ticket.

We remember that in [chapter 1, Configuring Your Own Zendesk](#), we decided the following:

- First of all, we need to remember, not all channels should be treated equally. This means that different channels will require different SLA rules. It is a common practice, for instance, that support requests created via Facebook are being escalated quicker than tickets created via the e-mail channel.
- Knowing that group 1 and group 2 have to be handled differently when it comes to escalation rules, we can take note that both groups have to be marked differently when the ticket is being created. Let's name each group. In this case, since we only have two different groups, we do not have to be too descriptive when naming them:

EscalationTypeA

EscalationTypeB

Reading this now, it may be a little confusing as we referred to the two different escalation types as groups, which in Zendesk terms they are not. It makes way more sense to use business rules to tag the tickets accordingly.

But why should we not use a custom ticket field?

While we could use a custom field, we should ask ourselves this: what would we gain by doing this?

We do not really need to visualize the escalation type of the ticket for our agents. On the contrary, we want the escalation workflows to happen automatically in the background. The only visible outcome should be the priority itself. Neat and tidy. There is also no need to manually change the escalation type of a ticket. Therefore, let's not create a ticket field for the escalation type and shift our attention to another area where it might be more useful.

ExampleComp sells its own software and most customer requests are probably related to one of its software products. It may be helpful for agents to know what software the request is related to without having to read the whole ticket description first.

It would also allow us to keep track of how many tickets we receive concerning each of ExampleComp's products.

So let's go ahead and add our first custom ticket field:

1. Click on the Admin icon (gear symbol) located at the bottom of Zendesk's sidebar.
2. Click on **Ticket Fields** located under **MANAGE** within the admin menu.
3. Click on **add custom field** located on the top-right within the main area:

The screenshot shows the Zendesk Admin interface. The left sidebar is highlighted with a red box and labeled '2'. The main content area is titled 'Ticket fields' and contains a table of 'Active fields'. The table has columns for 'Name', 'Type', and 'System'. The fields listed are: 'Betreff' (Text, System), 'Beschreibung' (Multi-line text, System), 'Status' (Drop-down, System), 'Typ' (Drop-down, System), 'Gruppe' (Drop-down, System), and 'Mitarbeiter' (Drop-down, System). In the top right corner of the main area, there is a red box highlighting the 'add custom field' button, labeled '3'.

Creating Custom Fields

You will then receive the option to choose one of the many different field types. Locate the **Drop-down list** field type and click on **select**:



Creating a custom ticket field differs from creating a custom user field. We can tell this straight away by the amount of options available to us:

The screenshot shows the configuration page for a new drop-down field. It includes sections for "For agents" and "For end-users". Under "For agents", there is a "Field title shown to agents" input field and a "Required" checkbox. Under "For end-users", there is a "Visible" checkbox checked, a "Title" input field, an "Editable" checkbox checked, and a "Description (optional)" input field. There is also a "Required" checkbox and a note about required values. The "Field options" section allows adding titles and tags, with a plus sign icon for "Add tag option". A note at the bottom says "Sort field options alphabetically by title upon save." A "Save" button is at the bottom right.

Drop-down field new

Hey! Want to categorize your drop-down options? [Learn more.](#)

For agents

Field title shown to agents

Required
Field cannot be blank when an agent solves a ticket.

For end-users

Visible
The field is visible to end-users on their ticket page.

Title

Field title shown to end-users

Editable
Field can be edited by the end-user when submitting a ticket online.

Description (optional)

Required
The end-user is required to enter a value for this field when submitting a ticket online.

Field options

On the ticket form users will see a drop-down field with the values you define in this section. The ticket will be tagged accordingly when submitted.

Title:

Tag:

Add tag option +

Sort field options alphabetically by title upon save.

Add field

Let's go through the process step by step:

1. First, we need to find a title as it is shown to our agents within the ticket view. We can keep it simple and go with Software. Next, we choose whether the drop-down can be blank when solving a ticket. In this case, we do not check the **Required** option as we may receive tickets that are not related to any of ExampleComp's software products:

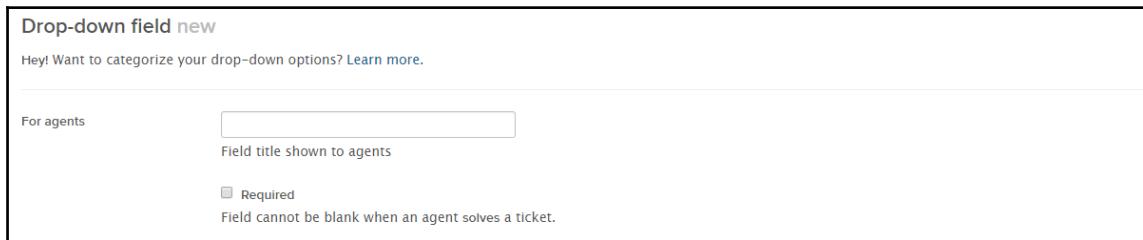
Drop-down field new

Hey! Want to categorize your drop-down options? [Learn more.](#)

For agents

Field title shown to agents

Required
Field cannot be blank when an agent solves a ticket.



2. Next, we choose whether we want our end-users to see this ticket field within their ticket page. We will tick the **Visible** box as we want our customers to select the right software when creating the ticket. Next, we choose a **Title** as shown to our customers. Again, we can simply go with Software.
3. In order to allow our end-users edit this field, we need to tick the **Editable** box. Adding a **Description** is optional but serves a great purpose. The same description will be visible in our support form and help our customers to understand what they are filling in.

Again, we will not check the **Required** box:

For end-users

Visible
The field is visible to end-users on their ticket page.

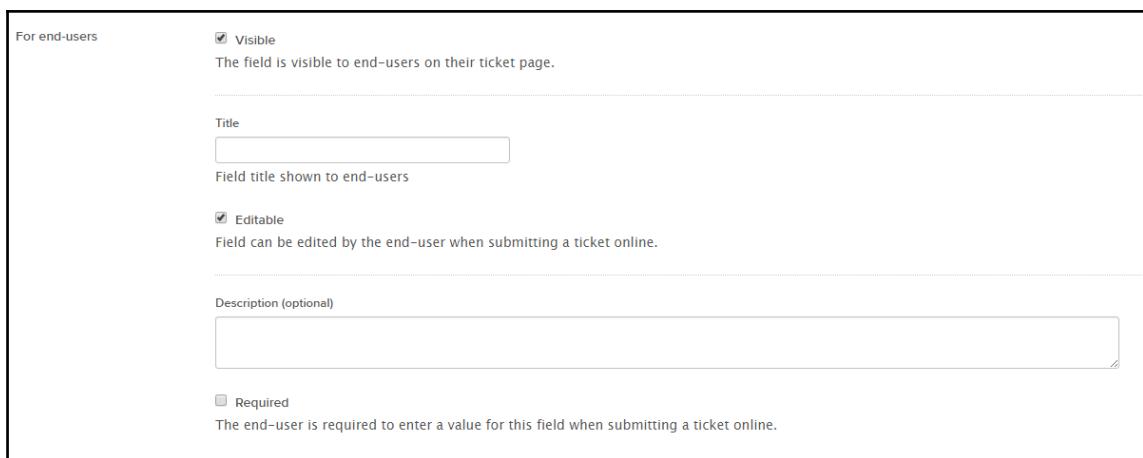
Title

Field title shown to end-users

Editable
Field can be edited by the end-user when submitting a ticket online.

Description (optional)

Required
The end-user is required to enter a value for this field when submitting a ticket online.



4. Now it is time to add each single selectable option. Each field option consists of a title and a corresponding tag. You may want to follow strict rules when it comes to naming your tags with a focus on readability.

An example would be dropdown_software_examplesoftware.

Once you are done adding all the software options, simply click on **Add field**:

Field options

On the ticket form users will see a drop-down field with the values you define in this section. The ticket will be tagged accordingly when submitted.

Title: -

Tag: +

Add tag option +

Sort field options alphabetically by title upon save.

Add field



Both for titles and descriptions, you are free to use dynamic content placeholders. That way, your support form can be displayed in different languages.

Custom organization fields

Adding custom organization fields is the same as adding user fields. Think about a way you could utilize a custom organization field and add it accordingly:

1. Click on the Admin icon (gear symbol) located at the bottom of Zendesk's sidebar.
2. Click on **Organization Fields** located under **MANAGE** within the admin menu.
3. Choose one of the field types on the right side and click on it.

Summary

In this chapter, you learned about fields in Zendesk. You learned about the general purpose of fields and how we can create and utilize the custom user, ticket, and organization fields. You now know their overall place in our Zendesk setup and what to keep in mind when planning and adding custom fields.

In the next chapter, you will learn about different ticket channels and how to set them up properly. We will use our ExampleComp road map to continue the journey of our Zendesk customization.

4

Setting Up Multiple Ticket Channels

Helping a customer the best way possible also means meeting your customer wherever they might be. The Internet provides many different *meeting points* and Zendesk surely kept up with the times by providing us with the necessary channels of communication.

However, it is important to know what channels make sense for you and your business. It is also important to keep in mind that the more channels you open, the more agents will be required to cater to the support needs.

In this chapter, we will take an in depth look at each channel provided by Zendesk. We will talk about their purpose, look at the implementation, and apply the newly gained insights to our ExampleComp scenario.

At the end of this chapter, you will know what channels to pick for your individual situation and how to set them up accordingly.

This chapter covers the following topics:

- What are channels?
- The available channels in Zendesk.
- Setting up e-mail, widget, social media, and Help Center channels.

What are channels?

Channels enable customers to engage with your agents. They allow customers to create tickets as well as establish an ongoing string of communication. Let's start by listing all the channels provided by Zendesk:

- Email
- Help Center
- Twitter
- Facebook
- Chat (Zopim)
- Voice
- Web Widget
- Mobile SDK
- API

While some channels are self-explanatory in their nature, understanding them properly goes a long way when it comes to the actual implementation. So let's look at them in detail.

Email channel

The **email** channel allows your customers to contact your support directly by writing to one or more designated e-mail addresses. As soon as an e-mail is received, it will be turned into a ticket within the Zendesk environment. Zendesk allows for an unlimited amount of e-mail addresses within your Zendesk domain.

An internal e-mail address created within your Zendesk domain would look something like this:

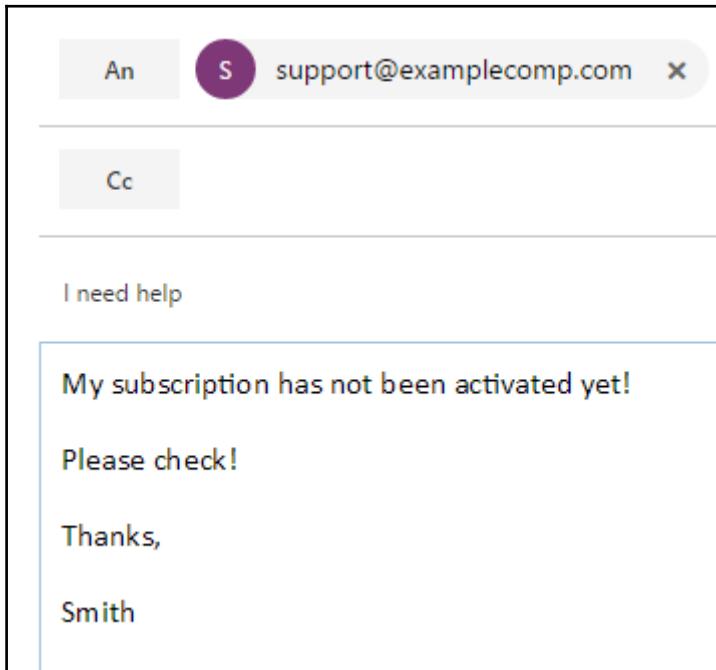
`support@mydomain.zendesk.com`

In our example, this would translate to the following:

`support@examplecomp.zendesk.com`

Obviously you would not want to supply internal e-mail addresses to the public and it makes sense to set up e-mail forwarding from an external e-mail address. In our example, that would look something like the following:

support@examplecomp.com forwarded to support@examplecomp.zendesk.com



This can be achieved by simply forwarding e-mails received at the external e-mail address to the internal address.

Zendesk e-mail channels also allow you to process outgoing e-mails as if they were sent using the external e-mail address. So, if an agent was to reply to a customer, the customer would see the external e-mail address as the sender.

Zendesk also allows for heavy customization when it comes to the look of your e-mail replies by giving the following two options:

- Customizing the look of the HTML template
- Customizing the look of the plain text template

It makes sense to customize both due to compatibility because not all customers' e-mail software will allow to display HTML.

Facebook channel

The Facebook channel allows customers to contact your support via Facebook. How does it work?

Zendesk allows you to add your company's Facebook page. It will then start monitoring it for timeline posts, comments, and private chat messages.

Zendesk also gives you the option to choose between the user interactions on your page, which you would like to turn into tickets. That makes sense as some companies may want their support to be private, allowing them to only turn private messages into tickets for their support.

For the end user, the communication would look something like this:



For the agent, while being aware that the ticket was created via the Facebook channel, it would look like most other tickets within their Zendesk environment. If the user decides to reply again, it will simply be added as a comment to the existing ticket.

Twitter channel

The Twitter channel allows you to screen and respond to Twitter messages. Similar to the Facebook channel, you can choose what tweets you would like to turn into tickets (also called twickets). Let's have a closer look at your options here:

- You receive the option to create Twitter searches in order to screen tweets related to your company

- You can choose between your Twitter searches and decide which ones should be turned into twickets
- Once you have converted a tweet into a ticket you can choose to reply via:
 - a tweet
 - a direct message
 - an e-mail
- You can set your own Twitter account as primary so that outgoing tweets seem to stem from your Twitter account

Just like for the Facebook channel, once a tweet has been turned into a ticket, the agent's workflow does not really differ.

On the other side, the customer feels like the support is provided straight via Twitter:



Chat (Zopim) channel

The Zopim Chat channel allows you to add a real-time chat option for your customers. After the chat is over, agents can turn the chat conversation into a ticket in order to update it later.

How can customers access the chat? You could, for instance, add Zopim to your website. However, many companies decide to add it to their companies Help Center.

Another popular application is providing targeted support by analyzing the user's browsing behavior. Zopim allows you to see what website visitors are up to in real time. For example, you could instruct agents to proactively approach customers who have already added a product to their shopping cart.

Talk channel

While a lot of customers prefer using the Internet to contact customer service, there is still a need for telephone support.

The Zendesk Talk channel allows you to add this option for your customers by picking a telephone number for incoming calls. This channel's features include:

- Setting call queue options
- Recording greetings
- Agents can take calls via Zendesk
- Calls are recorded and added to a ticket
- Customers can leave voicemails, which are automatically turned into tickets
- Phone forwarding

Why Zendesk Voice

Get started in minutes

We made it easy for anyone to set up phone support, right in Zendesk -- no additional software or hardware required. Choose a new Zendesk Voice number or port an existing US-based number.

Boost productivity

Voice is built into Zendesk, so you can manage phone support along with your other support channels. Resolve issues quickly with workflows and agent forwarding to analytics, customer details and knowledge base articles right at your fingertips.

Get smart about phone support

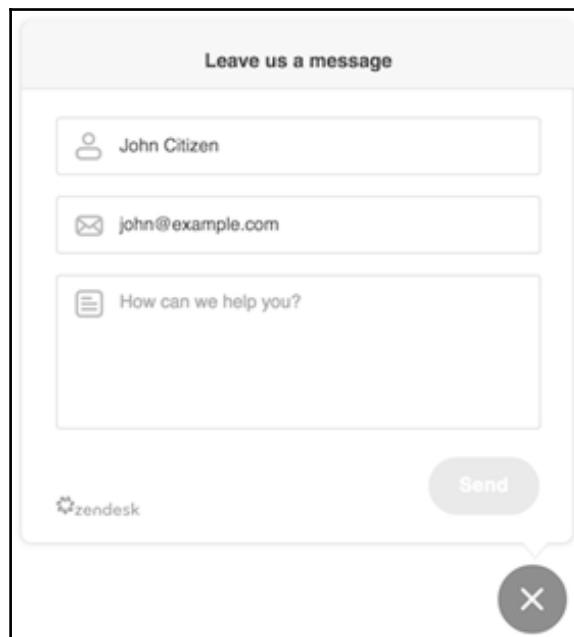
Zendesk Voice has all the features and reporting your support team needs, from call Know what's driving call volumes and make adjustments to keep your customers happy.

Web Widget channel

The Web Widget allows you to add a little widget to your website or Help Center giving your customers direct access to the following options:

- Support form to create a ticket
- Initiate live chat with support (if activated)
- Search the Help Center for appropriate articles

You can simply pick and choose your set of options and Zendesk will generate the necessary code snippet:



Mobile SDK channel

The Mobile SDK allows you to embed certain Zendesk options within a mobile app. This is especially great for app developers. You can give your customers quick access to the following:

- Submitting tickets
- Reviewing existing tickets
- Help Center access (without redirect)



API channel

Some of you might wonder: what is an API?

While we mentioned the Zendesk API before, let's make sure we fully understand what API means.

API stands for **Application Programming Interface**. An API consists of a set of functions, which developers can access for the creation of their own applications. Usually, in order to access data of the application, operating system, or other service that is providing the API.

How does that apply to Zendesk?

Accessing the Zendesk API means that you can use a range of Zendesk functions in your own code. One way to use the Zendesk API would be creating your own application for customers to create tickets.

The API channel page allows you to set up the necessary authentication needed in order to use the API.

A great amount of information can be found in Zendesk's API documentation:

https://developer.zendesk.com/rest_api/

The screenshot shows a web-based API documentation interface. At the top, a header reads "Core API". Below it is a sidebar menu with the following items:

- Core API (selected)
- Help Center API
- Zopim API
- Zendesk Voice API (with a cursor icon over it)
- API Clients
- Web Portal API
- Reseller API
- NPS API

At the bottom of the sidebar is a link labeled "Ticket Comments". To the right of the sidebar, the main content area has the following sections:

- Groups**: A brief description of what Groups are used for in Zendesk's ticket workflow.
- JSON Format**: A description of how Groups are represented in JSON.
- Table**: A table showing the properties of a Group object.

Name	Type	Read-only
id	integer	yes
url	string	yes

Setting up Zendesk channels

Now that we have a better understanding of Zendesk's channels, let's commence and go through the actual setups.

What channels did we plan to set up? Let's refer to our road map again:

- Email
- Facebook
- Twitter
- Widget
- Help Center / Support Form

Email channel setup

Let's start with the most common of channels: the Email channel.

Adding internal e-mail addresses

We will start by adding an internal e-mail address:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Email** located under **CHANNELS** within the admin menu.
3. Click on **Add address** in the right upper corner and then click on **Create new Zendesk address**:

Setting Up Multiple Ticket Channels

The screenshot shows the Zendesk Admin Home interface. On the left sidebar, under the 'CHANNELS' section, the 'Email' option is highlighted with a red box and the number '2'. The main content area is titled 'Channels / Email'. It displays several configuration options:

- Support addresses:** Zendesk supports an unlimited number of email addresses. Emails sent to any of these email addresses become tickets. A link to 'Learn more' is provided.
- Address:** A table with columns 'Address' and 'Name'. A red box labeled '3' highlights the 'Add address' dropdown menu, which includes options: 'Connect external address' and 'Create new Zendesk address' (also highlighted with a red box).
- Send email via Gmail:** A checkbox labeled 'Enable' is checked. A note states: 'Send responses via Gmail servers for authenticated Gmail support addresses. Deselect this if you see rate limit warnings from Google.'
- Accept wildcard emails:** A checkbox labeled 'Enable' is unchecked. A note states: 'Enables users to create tickets by sending email to any variation of your default Zendesk Support address: support@cfj.zendesk.com. That is, the "support" part can be anything.' A link to 'Learn more' is provided.
- Personalized email replies:** A checkbox labeled 'Enable' is checked. A note states: 'When enabled, the reply address includes the name of the agent or the end-user who is replying. For example, when an agent adds a public comment to a ticket, the notification email to the customer includes the name of the agent in the reply address.'

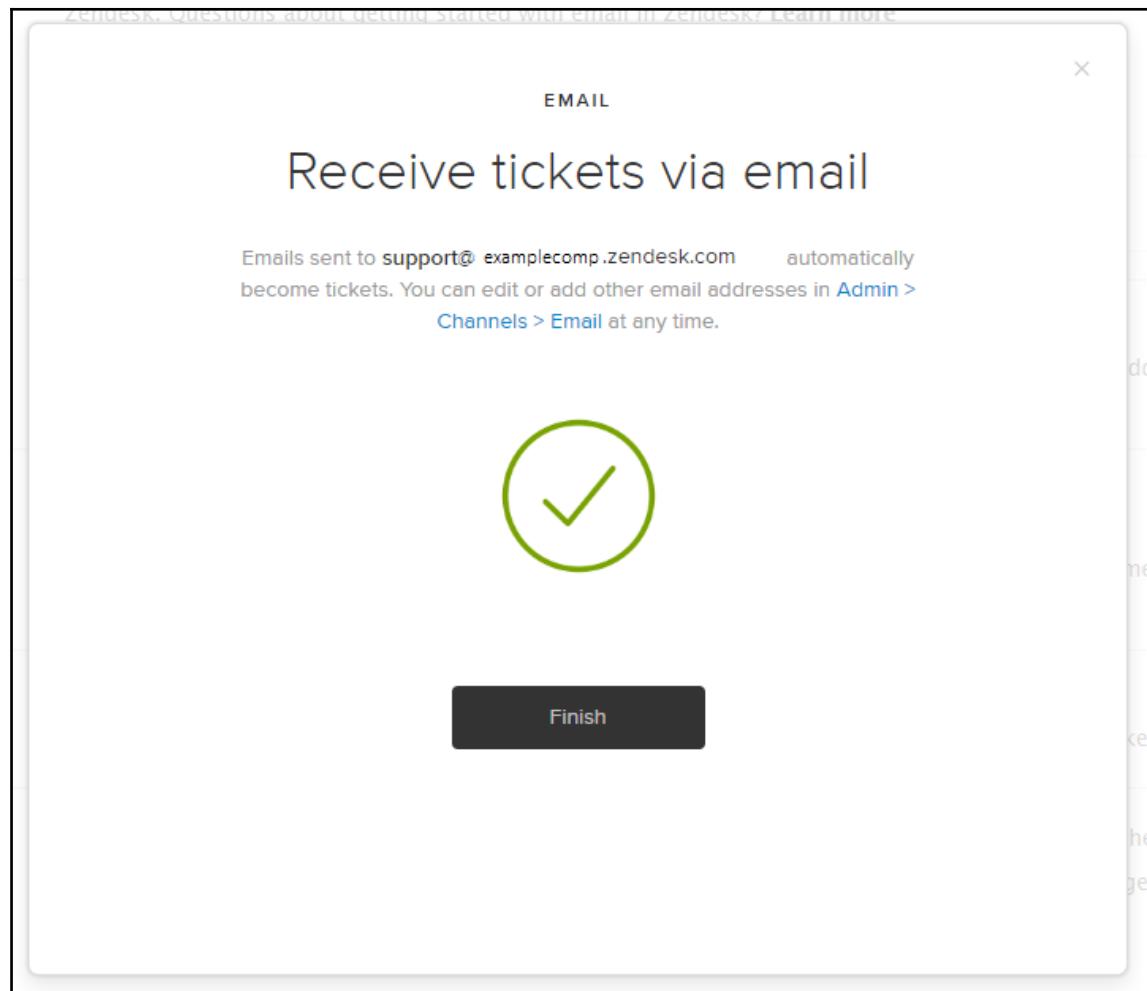
Next, we are asked to provide the first part of the e-mail address. As this will be our main support address, let's go with support:

The dialog box is titled 'Create a new support address'. It contains the following text:
Any email sent to this address will show up as a ticket in your Zendesk

A text input field contains the text 'support' with a green border, followed by '@ examplecomp.zendesk.com'. Below the input field, a suggestion list shows: 'Suggestions: info, help, sales, contact, jobs, hello'

A large 'Create now' button is at the bottom of the dialog.

Once we click on **Create now**, the new internal e-mail address is ready for us. E-mails sent to support@examplecomp.zendesk.com automatically become tickets:



Our newly created internal e-mail address is now listed on the same page we navigated to in order to create it. You may come back to this page any time you feel the need to edit or to create a new e-mail address.

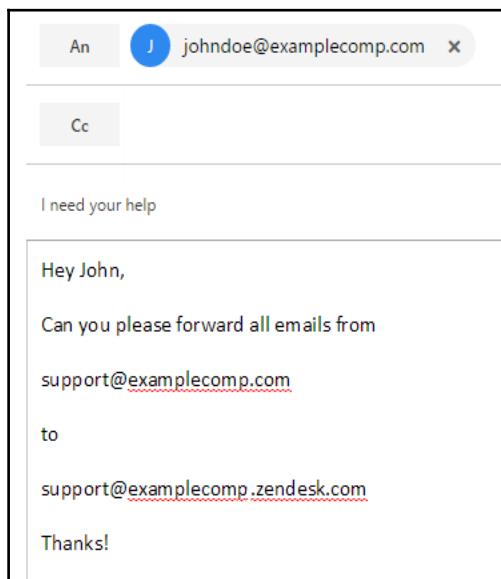
The screenshot shows the 'Channels / Email' section of the Zendesk admin interface. Under 'Support addresses', it says 'Zendesk supports an unlimited number of email addresses. Emails sent to any of these email addresses will show up as tickets in your Zendesk. Questions about getting started with email in Zendesk? Learn more'. A table lists one address: support@examplecomp.zendesk.com (labeled as a 'Zendesk address') next to ExampleComp. There are 'Add address' and 'edit' buttons.

Obviously, we still need to make sure that our customer's e-mails find their way to us.

Forwarding example with Outlook.com

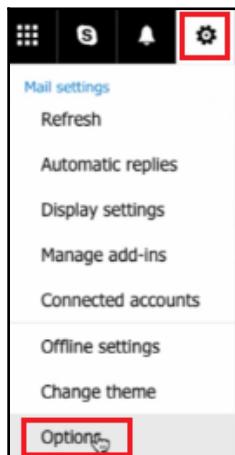
While most of us might complete this step simply by sending an e-mail to the right department, let's go through one quick example anyway.

So, you can choose not to do this:

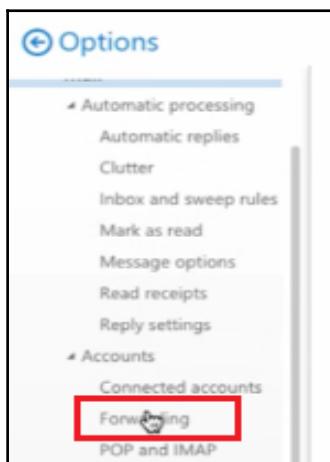


Alternatively, if you have the login details, you could do the following steps:

1. Log in to Outlook.com using the credentials of the e-mail that you would like to use as your external address (`support@examplecomp.com`).
2. Move your mouse cursor over the gear icon in the upper right corner and proceed by clicking on **Options**:

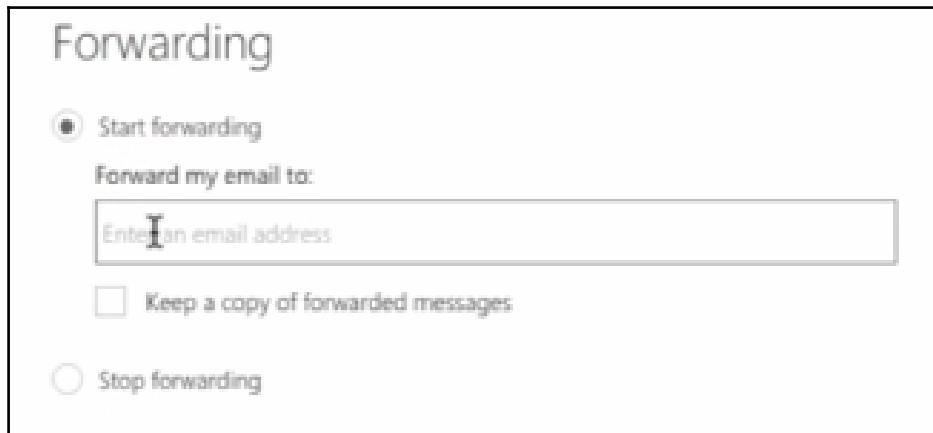


3. First we need to open the **Options** menu.
4. Under **Mail | Accounts**, click on **Forwarding**:



Next we look for the **Forwarding** settings:

1. Click on **Start forwarding** and enter your internal Zendesk e-mail address (`support@examplecomp.zendesk.com`).
2. Click on **Save**:



3. Finally, we enter our internal e-mail address.

If your company is not using Outlook.com, you may want to do a quick online search in order to find out how to set up forwarding. The idea behind it remains the same.

Great! Our customers can officially create tickets within our Zendesk environment by sending an e-mail to the ExampleComp e-mail address.

But what happens when we reply? Will the user receive the reply from the internal or the external e-mail address?

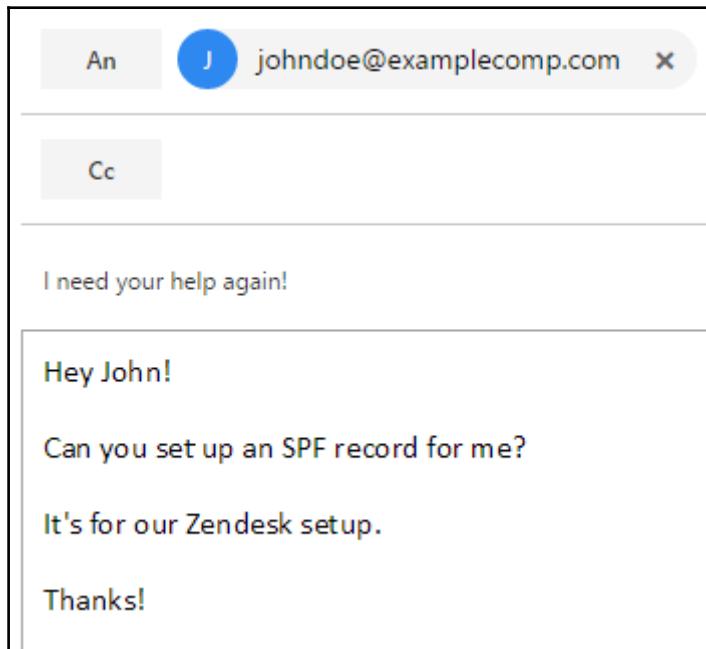
Good question! In our current setup, Zendesk does not have the necessary rights to act as if they are replying from our external e-mail address. Instead, our customer will receive a reply from `support@examplecomp.zendesk.com`.

That is not ideal. We might want to set up a **Sender Policy Framework (SPF)** record in order to allow Zendesk to send e-mails on behalf of our ExampleComp e-mail server.

Create SPF record

Again, it would probably make sense to get someone else to do this for us. However, it is always great to know how things actually work and how they are accomplished.

So, you can choose not to do this:



Alternatively, you could try doing it yourself.

While the core idea does remain the same, the actual step-by-step process depends on your domain registrar.

What we need to do is to create an SPF record to reference Zendesk. This entry will tell our domain that it is fine for Zendesk to send e-mails on its behalf. This can be achieved by navigating to our domain's **Domain Name System (DNS)** settings.

Let's assume that ExampleComp used `namecheap.com` for their domain:

The screenshot shows the Namecheap Domain List interface. On the left, there is a sidebar with icons for Dashboard, Expiring Soon, Domain List (which is highlighted with a red box), Product List, and Profile. The main area is titled "Domain List" with a "REFRESH" button. It has filters for Actions, Filters, and Search. Below that are columns for All, Products, and Expiration. A single domain entry is listed: "examplecomp.com" with a house icon, a dropdown arrow, and a "MANAGE" button.

This is what our domain list would look like on `namecheap.com`:

1. Log in to your Namecheap account.
2. Click on **Domain List** located in the left-side menu.
3. Click on **MANAGE** next to the domain (`examplecomp.com`).
4. Click on the **Advanced DNS** tab.
5. Click on the **Add new record**.

You will be prompted to supply three bits of information:

- Type
- Value
- Host

Simply select **TXT Record** as the **Type**, choose **@** as **Host**, and the following text as the **Value**:

```
v=spf1 include:mail.zendesk.com ?all
```

Make sure to submit your new entry by clicking on **Save all changes!**

That is it! Most of you might not need to do this individually, but having to send an e-mail to another department is so much easier when we understand the subject a little better.

Email channel settings

Let's have a look at each setting provided by Zendesk when it comes to our Email channel:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Email** located under **Channels** within the admin menu.

Accept wildcard emails

The first option is called **Accept wildcard emails**:

Accept wildcard emails **Enable**
Enables users to create tickets by sending email to any variation of your default Zendesk support address: support@`examplecomp.zendesk.com`. That is, the 'support' part can be anything. Learn more

When this option is enabled, customers can contact us by sending an e-mail to any variation of our default Zendesk address. So if a customer was to send an e-mail to `somerandomword@examplecomp.zendesk.com`, we would still receive it.

In our scenario, we will deactivate this option.

Personalized email replies

The **Personalized email replies** option allows us to use personalized e-mail addresses when responding to a customer:

Personalized email replies **Enable**
When enabled, the reply address includes the name of the agent or the end-user who is replying.
For example, when an agent adds a public comment to a ticket, the notification email to the customer includes the name of the agent in the reply address.

When this option is enabled, the agent's name would be added to the e-mail address.

In our scenario, we will deactivate this option also.

Gmail Go-to Actions

The **Gmail Go-to Actions** have been around for a while now, and in case you are using Google's mail service, you may have noticed them before:

Gmail Go-to Actions **Enable**
When enabled, agents and end-users will have buttons in their Gmail inbox that enable them to take quick actions, without opening emails.

This option would add quick actions to the end-user's inbox. Such quick actions only work for Gmail accounts.

For our example, we do not need to enable this option either.

Email templates

The name **Email templates** can easily be mistaken for predefined answers to our customers. Those, however, are called macros. In this case, we are talking about the look of our outgoing e-mails instead:

The screenshot shows the 'Email templates' configuration page. It includes sections for 'HTML template' and 'Text template' with their respective code editors, a checkbox for showing user profile photos, and revert-to-default buttons.

HTML template:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
2 <html>
3 <head>
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5   <style type="text/css">
6     table td {
7       border-collapse: collapse;
8     }
9     {{styles}}
10  </style>
11 </head>
12 <body {{attributes}} style="width: 100%; margin: 0; padding: 0;">
13   <div style="padding: 10px; line-height: 18px; font-family: 'Lucida Grande', Verdana, Arial, sans-serif; font-size: 12px; color: #444444;">
14     <div style="color: #8b8565;">{{delimiter}}</div>
15     {{content}}
16   </div>
17   <div style="padding: 10px; line-height: 18px; font-family: 'Lucida Grande', Verdana, Arial, sans-serif; font-size: 12px; color: #aaaaaa;">
18     {{footer}} {{footer_link}}
19   </div>
20 </body>
21 </html>
22
23
```

Text template:

```
1 {{content}}
2
3 {{footer}}
```

We remember the following from *Email channel* section:

Zendesk also allows for heavy customization when it comes to the look of your e-mail replies by giving the following two options:

Customizing the look of the HTML template

Customizing the look of the plain text template

Being able to customize the look of our e-mails allows us to adjust the look according to our brand. We can also make sure to include all the necessary information in our e-mail footer.

Mail delimiter

Mail delimiter is a line of text, letting our customer know that their response should be entered above this very same line of text:

Mail delimiter

Between 20 and 65 characters.

The mail delimiter contains a line of text that informs the email recipient that any text entered into a reply must be above a certain line in the email. This is used to prevent all of the other text in the email from being added to the reply (the comment that is added to the ticket). Enter {{txt.email.delimiter}} if you want the delimiter text displayed in the language selected in the email recipient's user profile Language setting.

In this case, we are using the standard dynamic content placeholder to make sure we get every language covered:

```
{ {txt.email.delimiter} }
```

In English, this will translate to the following:

```
# #- Please type your reply above this line -# #
```

Custom domain for DKIM

Custom domain for DKIM could be defined as a security option:

Custom domain for DKIM Enable

To use the domain of your support addresses as the DKIM domain, add two CNAME records under the domain of your support addresses for the selectors zendesk1 and zendesk2. For example:

1. zendesk1._domainkey.example.com must point to zendesk1._domainkey.zendesk.com
2. zendesk2._domainkey.example.com must point to zendesk2._domainkey.zendesk.com

Unfortunately, nowadays, it has become quite common and easy to spoof e-mails. You may have received such an e-mail before.

Someone you know sends you an e-mail that clearly does not come from them personally. This is called spoofing: sending e-mails with a forged sender address.

In order to fight this, you can digitally sign your e-mails with **Domain Keys Identified Mail (DKIM)**. While this will not prevent spoofing, it allows you to prove that you, the e-mail sender, are authorized.

Enabling this option is not enough. Again, we will need to make some changes to our DNS records. While the core idea remains the same, the actual step-by-step process depends on your domain registrar.

This time we will add the following two **CNAME** records:

1. `zendesk1._domainkey.example.com` must point to `zendesk1._domainkey.zendesk.com`
2. `zendesk2._domainkey.example.com` must point to `zendesk2._domainkey.zendesk.com`

Facebook channel setup

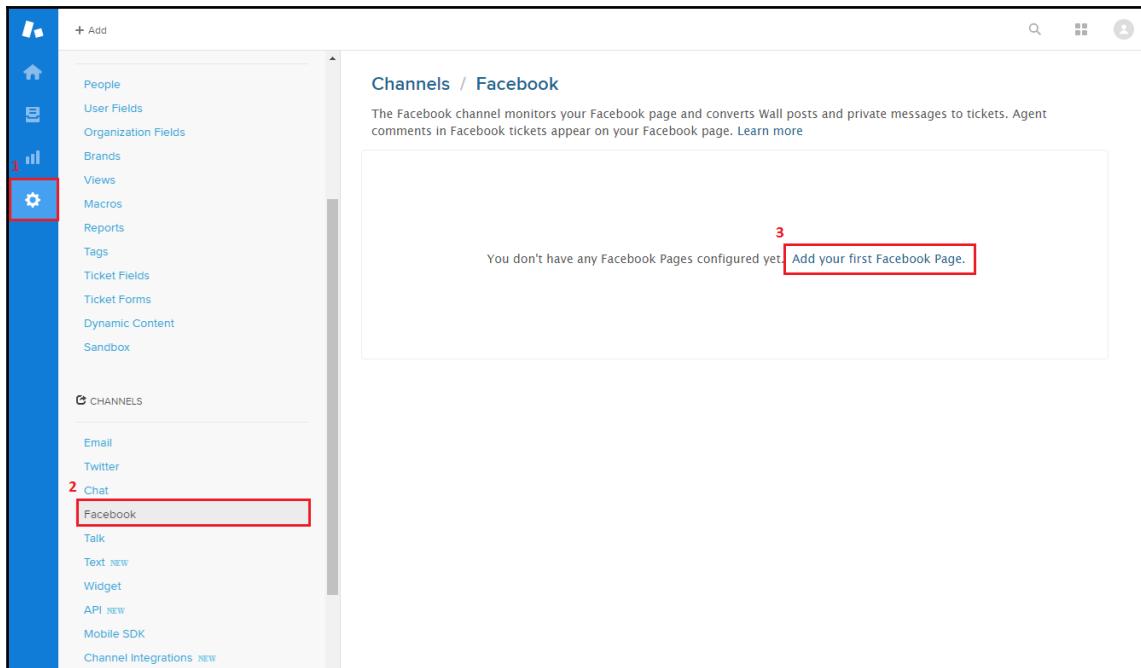
Let's move on to the Facebook channel. Nowadays a lot of customers turn to the Facebook pages of companies expecting a faster and more accurate response than the public forum.

Adding your first Facebook page

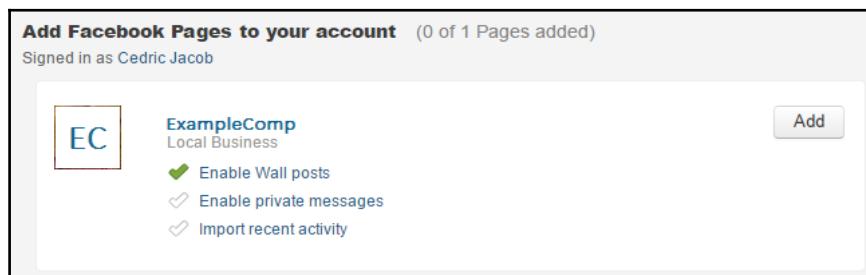
We will start by adding our first Facebook page:

1. Click on the Admin icon (gear symbol) located at the bottom of Zendesk's sidebar.
2. Click on **Facebook** located under **Channels** within the admin menu.

3. Click on **Add your first Facebook Page:**



If you have not yet logged into Facebook, you will be prompted to log in now. As soon as you have logged in, a list of the pages that you manage with your account shows up:



There are three options to look at before clicking on **Add**:

- **Enable Wall posts**
- **Enable private messages**
- **Import recent activity**

Enable Wall post will turn all future Wall posts into tickets while **Enable private messages** will turn private messages sent to your page into tickets. We will tick both boxes for our scenario.

Ticking the box next to **Import recent activity** would lead to Zendesk importing all recent Facebook Wall posts and private messages, thereby turning them into tickets retroactively. We do not want this for now.

After clicking on **Add**, we are presented with a list of all our Facebook pages connected to our Zendesk environment:

The screenshot shows the 'Channels / Facebook' section. It displays a single connected Facebook page named 'ExampleComp'. The page is categorized as 'Local Business'. There are buttons for 'add new Page' and 'edit'. A note at the top states: 'The Facebook channel monitors your Facebook page and converts Wall posts and private messages to tickets. Agent comments in Facebook tickets appear on your Facebook page. Learn more'.

This is really all we had to do in order to add our Facebook channel. Let's have a look at our Facebook channel settings next.

Facebook channel settings

There are not too many settings for us to play with, resulting in a rather sparse looking settings menu:

The screenshot shows the 'Facebook channel settings' menu. It includes two main sections: 'Include Wall posts' and 'Include private messages'. Under 'Include Wall posts', there is a radio button for 'Yes' (selected) and 'No'. A note says: 'Automatically convert any Wall post on the Page to a ticket. Agents responses on the ticket will appear on the Facebook Wall.' Below this is a checked checkbox for 'Include Wall posts authored by the Page', with a note: 'Choose to capture Wall posts authored by this Facebook Page along with other Facebook users.' Under 'Include private messages', there is a radio button for 'Yes' (selected) and 'No'. A note says: 'Convert Facebook Page messages to tickets and have a private conversation with your customers.' Below this is a note: 'If your Page becomes unauthorized, you may need to add the 'Read Mailbox' permission for this feature to work. [Add permission now!](#)' At the bottom right is a 'Update Page settings' button.

All the settings are self-explanatory and have already been covered when setting up the channel:

- **Include Wall posts**
- **Include private messages**

Twitter channel setup

Adding a Twitter channel can be a bit more complex than adding a Facebook channel. This is because of the amount of options we get when working with Twitter accounts. From the Twitter channel section, we remember the following:

Let's have a closer look at your options here:

You receive the option to create Twitter searches in order to screen tweets related to your company.

You can choose between your Twitter searches and decide which ones should be turned into twickets.

Once you have converted a tweet into a ticket you can choose to reply via...

tweet

direct message

email

You can set your own Twitter account as primary so that outgoing tweets seem to be sent from your Twitter account.

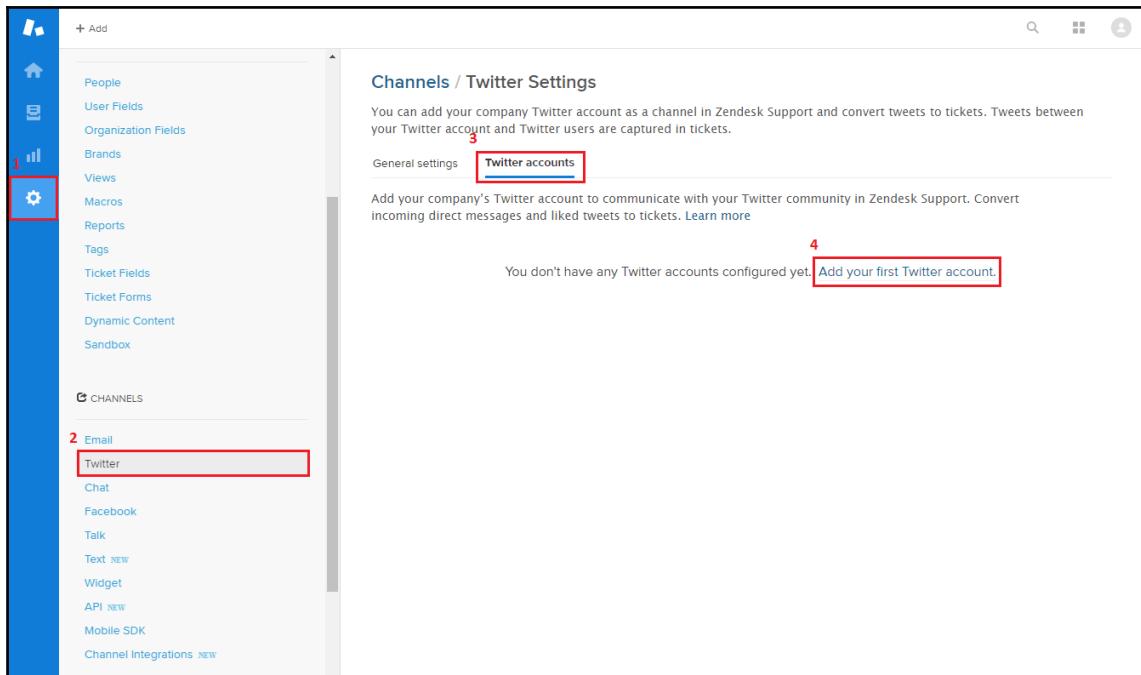
Before we can take advantage of all these options, let's add our first Twitter account.

Adding your first Twitter account

We will start by adding our first Twitter account:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Twitter** located under **CHANNELS** within the admin menu.
3. Click on the **Twitter Accounts** tab.

4. Click on **Add your first Twitter account**:



If you have not yet logged into Twitter, you will be prompted to log in now. As soon as you have logged in, the account will be added to Zendesk:



Twitter channel settings

When it comes to the Twitter channel, there are two kinds of settings:

- General settings
- Account related settings

While the **General** settings apply to all Twitter accounts, account related settings only apply to one specific account:

The screenshot shows the 'Channels / Twitter Settings' page. At the top, there are three tabs: 'General settings' (highlighted with a red box), 'Twitter accounts' (selected), and 'Manage searches'. Below the tabs, a sub-header reads: 'Add your company's Twitter account to communicate with your Twitter community in Zendesk. Zendesk can convert incoming direct messages and liked tweets to tickets. Tweets between your Twitter account and Twitter users are captured in tickets. [Learn more](#)'. Under 'Active Twitter accounts', there is a list with one item: '@ exampleComp'. To the right of this item is a red box containing the number '2' and the word 'edit'.

We will have a look at both the general settings and the account related settings.

Account-related settings

After clicking on **edit** next to our ExampleComp Twitter account, we receive the following options, all of which can be answered with a simple **Yes** or **No**:

- **Allow replies via this account**
- **Capture public mentions as tickets**
- **Capture incoming direct messages as tickets**
- **Track likes**

The screenshot shows the 'Twitter Settings' page. It contains four sections with configuration options:

- Allow replies via this account:** Radio buttons for 'Yes' (unchecked) and 'No' (checked). A note below says: 'Allow agents to reply from this Twitter account when responding to a ticket.'
- Capture public mentions as tickets:** Radio buttons for 'Yes' (checked) and 'No' (unchecked). A note below says: 'Automatically convert any public tweet containing your Twitter account username (for example, @zendesk) to a ticket.'
- Capture incoming direct messages as tickets:** Radio buttons for 'Yes' (unchecked) and 'No' (checked). A note below says: 'Automatically convert incoming direct messages into tickets and allow customers to reach you privately.'
- Track likes:** Radio buttons for 'Yes' (unchecked) and 'No' (checked). A note below says: 'Configure your Twitter account to automatically convert liked tweets to tickets. Many popular Twitter clients have the ability to mark a tweet as liked.'

At the bottom right are two buttons: 'Reauthorize or' and 'Update Twitter account' (highlighted with a red box).

Let's review each option on its own:

- **Allow replies via this account:**

Enabling this option will allow agents to reply from this Twitter account when responding to a ticket. As we intend to do this, we will choose **Yes**.

- **Capture public mentions as tickets:**

This allows us to convert any public tweet that contains our username (@ExampleComp) into a ticket. In our case, we will enable this option.

- **Capture incoming direct messages as tickets:**

Enabling this option will turn direct messages into tickets. Let's tick **Yes** and enable this option as well.

- **Track likes:**

This option allows us to automatically convert liked tweets to tickets. We will keep this option disabled.

General settings

Next, we will have a look at the **General settings**:

The screenshot shows the 'Channels / Twitter Settings' page. At the top, there is a brief description: 'You can add your company Twitter account as a support channel in your Zendesk and convert tweets to tickets. Tweets between your Twitter account and Twitter users are captured in tickets.' Below this, there are three tabs: 'General settings' (which is selected), 'Twitter accounts', and 'Manage searches'. A note below the tabs says: 'Configure settings for outgoing tweets in your Twitter channel. [Learn more](#)'.

The 'General settings' section contains two main configuration items:

- Enable twitter search?**: There are two radio buttons: Yes and No. A note next to it says: 'Enabling Twitter search adds the Twitter saved searches menu to your Zendesk. All administrators have access to this menu. You can allow agents access to it by granting them Twitter access permission in [roles](#)'.
- Append ticket links to outgoing tweets?**: This section includes a note: 'In order to activate this feature you must enable Twitter login for your customers. Enable'. It also states: 'When responding to a customer on Twitter, provide a shortened ticket URL in your tweet. This allows your customers to leave a longer response than is possible in a tweet.' and 'Appending a ticket URL to outgoing tweets allows customers to respond directly in the ticket and provide more detail than is allowed via Twitter.'

In the bottom right corner of the form, there is a 'Save tab' button.

The following two options are available:

- **Enable twitter search?**
- **Append ticket links to outgoing tweets?**

In order to understand option 1, we will need to cover another tab called **Manage searches**. Before we do that, let's get option 2 out of the way.

So do we want to append ticket links to outgoing tweets?

When enabled, Zendesk will add a shortened ticket URL in our tweet, which will allow the customer to leave a longer response than usual. This might be helpful due to tweets being limited to a small amount of characters. In order for this to work, we need to enable Twitter login for our customers (SSO). As we did not plan to provide such a feature, we will leave it out for now.

So what about the first option again? Should we enable Twitter search?

Enabling this feature allows agents to access saved Twitter searches. Once they find a tweet that they might want to interact with, they can turn it into a *twicket*. In order for this to work, we will need to do two things:

1. Give agents the Twitter access permission by editing their role.
2. Create a Twitter search for our agents to access.

We already know how to do the first. So let's focus on step two.

Manage searches

Managing searches allows us to create Twitter searches based on keywords. We simply need to click on the **Manage searches** tab, followed by the **add search** button:

The screenshot shows the 'Channels / Twitter Settings' page. At the top, there is a note: 'You can add your company Twitter account as a support channel in your Zendesk and convert tweets to tickets. Tweets between your Twitter account and Twitter users are captured in tickets.' Below this, there are three tabs: 'General settings', 'Twitter accounts', and 'Manage searches'. The 'Manage searches' tab is highlighted with a red box and the number '1' above it. A sub-section explains: 'A saved search is a Twitter search based on predefined keywords. Viewing a saved search in Zendesk returns a stream of tweets that match the keywords defined in the saved search.' It also states: 'Agents can select tweets to convert to tickets and reply to customers via Twitter directly from the saved search stream. Learn more'. Below this, a table lists saved searches. The first row, 'Shared', has a red box around the 'edit' button and the number '2' above it. The second row, 'test', has a red box around the 'Reorder' button.

Shared	edit
test	Reorder

Creating a Twitter search consisting of the following parts:

- The name of the search entry
- A reference to which Twitter account we want to use when responding
- The keywords that Zendesk is basing the Twitter search on
- The groups that have access to this Twitter search

In Zendesk, this translates into the following:

- **Name your search**
- **Respond to tweets as**
- **Keywords**
- **Search available for**

Twitter Settings

Name your search

Respond to tweets as The default Twitter account for responding to tweets in this search.

Keywords Zendesk will search Twitter for tweets that match these keywords. Click "Preview Twitter Search" to test your keywords.

Search available for All agents
 Agents in group
 Me only

[Preview Twitter Search](#) Live search results based on the keywords you have entered above.

What can we achieve by doing this?

We can create a search using our ExampleComp product names as keywords and have agents review those searches whenever their other ticket views are empty.

It makes sense in our scenario. So go ahead and set it up.

Widget channel setup

Next up, we have the Widget channel. Many of us come across these types of widgets on a daily basis. You could say, they became a new common way of interacting with customers.

About Zendesk's widget

The widget channel is great for offering support where the customer might need it the most. We can add the widget to any page of our website and assure that less customers leave the page to find answers first.

Setting up the widget can be divided into two steps:

1. Creating the code snippet.
2. Adding the code to your website's page.

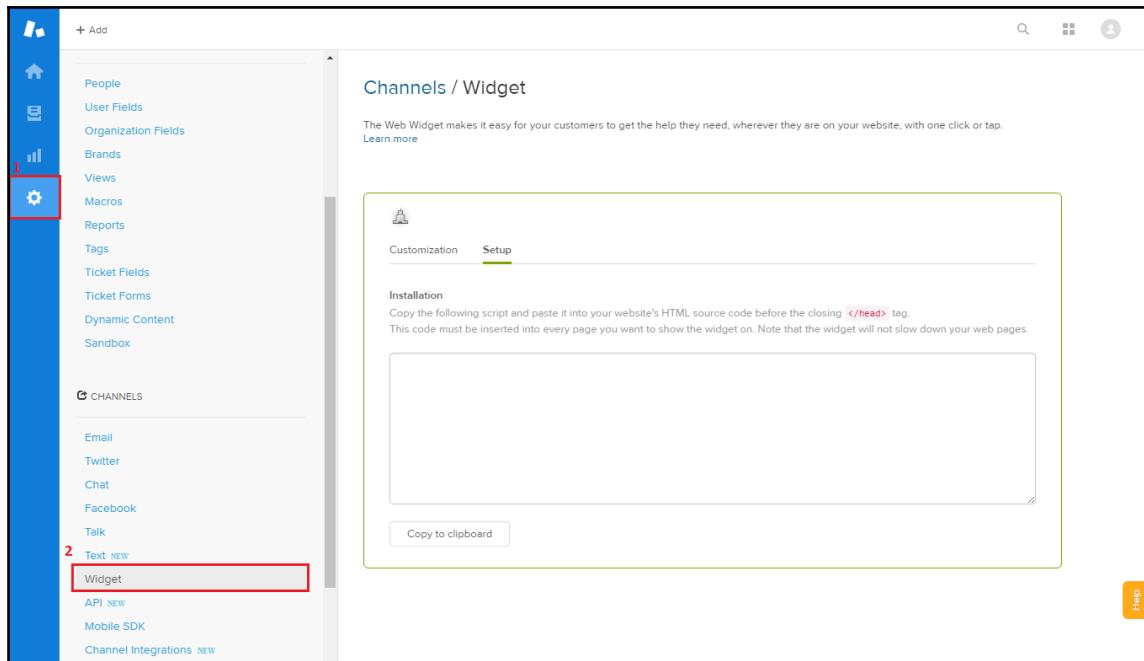
First, we will concentrate on step 1.

Widget customization

So let's go ahead and navigate to the corresponding setup page:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.

2. Click on Widget located under CHANNELS within the admin menu:



Next, we will click on the **Customization** tab. Zendesk will then proceed and present to us all the available customization options. Before going through them step by step, let's list them first:

- **Contact form** (On/Off)
- **Custom tickets fields** (Drop-down)
- **Chat** (On/Off, Reference to Chat setup page)
- **Help Center** (On/Off)

- **Zendesk logo** (On/Off)
- **Theme color** (Color selector)
- **Position** (Drop-down)
- **Web Widget button text** (Text Field)
- **Contact form button text** (Drop-down)

Customization Setup

Contact form
A simple contact form that enables users to easily get in touch with you, creating a ticket in your Zendesk.

Custom ticket fields
Select up to two custom ticket fields to include in the contact form of your widget.

Custom ticket fields 

[Settings](#)

Chat
Chat directly with your customers via your website using Zopim, Zendesk's live chat product.

[Set up Chat](#)

Help Center
Deflect tickets by serving knowledge base articles.

[Settings](#)

Zendesk logo
Show Zendesk logo in your widget.

Theme color
Select a custom color for buttons and links in your widget.



Position
Select left or right position of the Web Widget.

Right 

Web Widget button text
The text displayed on the Web Widget button (not shown on mobile).

Help 

Contact form button text
The text displayed on the button that launches the contact form.

Leave us a message 

[Save](#)

The **Contact form** option will add a simple contact form to the widget, allowing customers to create a ticket on the fly.

The **Custom ticket fields** option allows us to add any customer fields that we have created earlier to the contact form. We added custom fields in Chapter 3, *Creating Custom Fields* where we discussed the following:

ExampleComp sells their own software and most customer requests are probably related to one of their software products. It may be helpful for agents to know what software the request is related to without having to read the whole ticket description first.

So in our case, we might want to show some of those fields.

The **Chat** option allows us to add a chat function to our widget. As we did not set up a chat channel yet, we see a reference to the setup page instead. Let's keep this in mind. We might decide to add the chat channel later.

The **Help Center** allows us to deflect tickets by allowing the user to self-serve by looking at any articles we offer in our **Help Center**. Again, while we have not created a **Help Centre** yet, we might want to keep this option in mind.

The **Zendesk logo** option allows us to display the Zendesk logo. Why not? Zendesk is great!

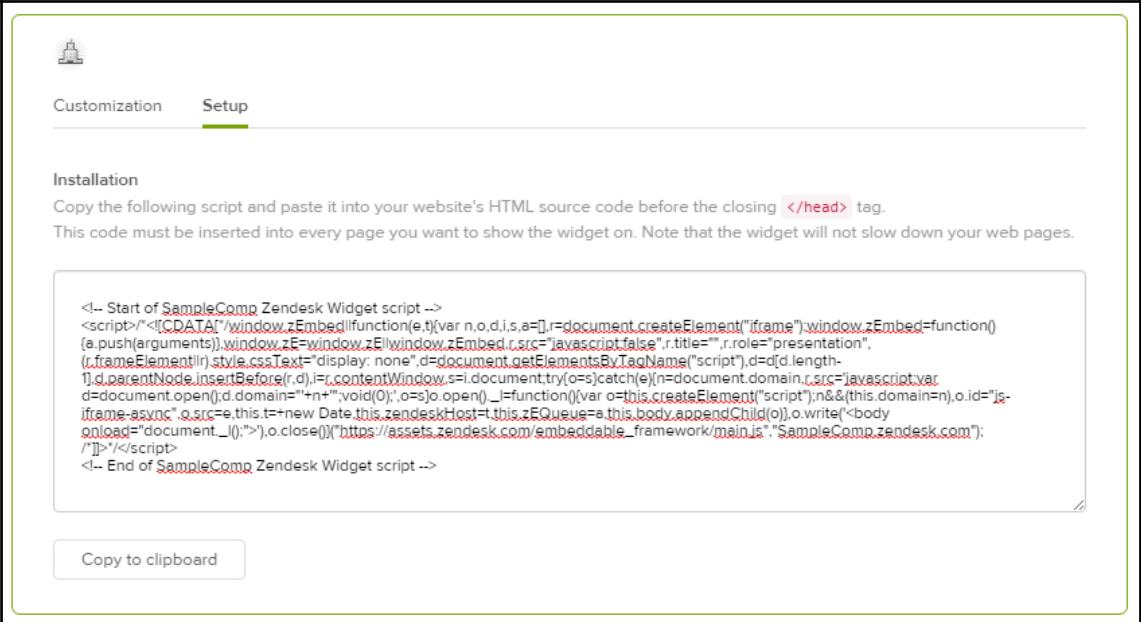
Up next, we find the settings regarding the look and the position of our widget. Let's adapt the **Theme color** according to our brand, pick **Right** as the **Position**, and choose **Support** as the **Web Widget button text**. Last but not least, for the **Contact form button text** the simple **Contact us** solution should suffice.

All we need to do now is click on the Save button and review the generated snippet code.

Widget snippet code

Zendesk will automatically generate the necessary code that can be copied into our HTML source code. It is messy in its original state, which does not stop us or the developers from tidying it up later on. We might want to forward this to our developers:

```
<!-- Start of SampleComp Zendesk Widget script -->
<script>/*<![CDATA[*/window.zEmbed||function(e,t){var
n,o,d,i,s,a=[],r=document.createElement("iframe");window.zEmbed=function(){
a.push(arguments)},window.zE=window.zE||window.zEmbed,r.src="javascript:fal
se",r.,r.role="presentation", (r.frameElement||r).style.cssText="display:
none",d=document.getElementsByTagName("script"),d=d[d.length-1],d.parentNod
e.insertBefore(r,d),i=r.contentWindow,s=i.document;try{o=s}catch(e){n=docum
ent.domain,r.src='javascript:var
d=document.open();d.domain="'+n+'";void(0);',o=s}o.open()._l=function(){var
o=this.createElement("script");n&&(this.domain=n),o.id="js-iframe-
async",o.src=e,this.t+=new
Date>this.zendeskHost=t>this.zEQueue=a>this.body.appendChild(o),o.write('<
body
onload="document._l();">'),o.close()}("https://assets.zendesk.com/embeddabl
e_framework/main.js","SampleComp.zendesk.com");
/*]]>*</script>
<!-- End of SampleComp Zendesk Widget script -->
```



The screenshot shows the Zendesk Widget configuration interface. At the top, there are two tabs: "Customization" and "Setup". The "Setup" tab is currently selected. Below the tabs, under the heading "Installation", there is a text area containing the Zendesk widget script. The entire script is highlighted in red. At the bottom left of this text area, there is a button labeled "Copy to clipboard".

```
<!-- Start of SampleComp Zendesk Widget script -->
<script>/*<![CDATA[*/window.zEmbed||function(e,t){var
n,o,d,i,s,a=[],r=document.createElement("iframe");window.zEmbed=function(){
a.push(arguments)},window.zE=window.zE||window.zEmbed,r.src="javascript:false",r.title="",
r.role="presentation",
(r.frameElement||r).style.cssText="display: none",d=document.getElementsByTagName("script"),d=d[d.length-1].parentNode.insertBefore(r,d),i=r.contentWindow,s=i.document;try{o=s}catch(e){n=document.domain,r.src='javascript:var
d=document.open();d.domain="'+n+'";void(0);',o=s}o.open()._l=function(){var
o=this.createElement("script");n&&(this.domain=n),o.id="js-iframe-
async",o.src=e,this.t+=new Date>this.zendeskHost=t>this.zEQueue=a>this.body.appendChild(o),o.write('<
body
onload="document._l();">'),o.close()}("https://assets.zendesk.com/embeddable_framework/main.js","SampleComp.zendesk.com");
/*]]>*</script>
<!-- End of SampleComp Zendesk Widget script -->
```

Help Centre (Support Form) channel setup

The Help Centre channel differs significantly from the other channels. First of all, and you may have noticed already, that there is no setup page for it. This is due to the fact that it has to be set up in a different way altogether.

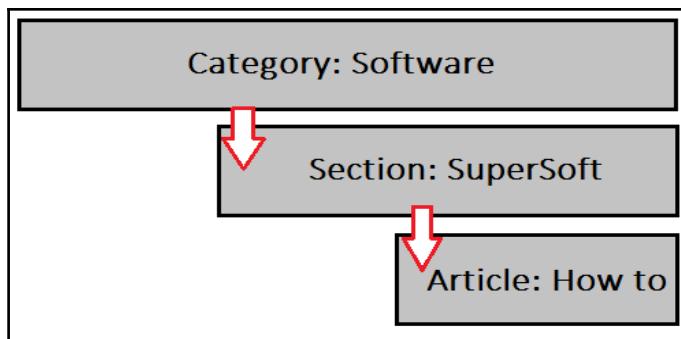
Let's start by gaining a little more insight first.

Zendesk's Help Centre— A quick overview

So what is Zendesk's Help Centre?

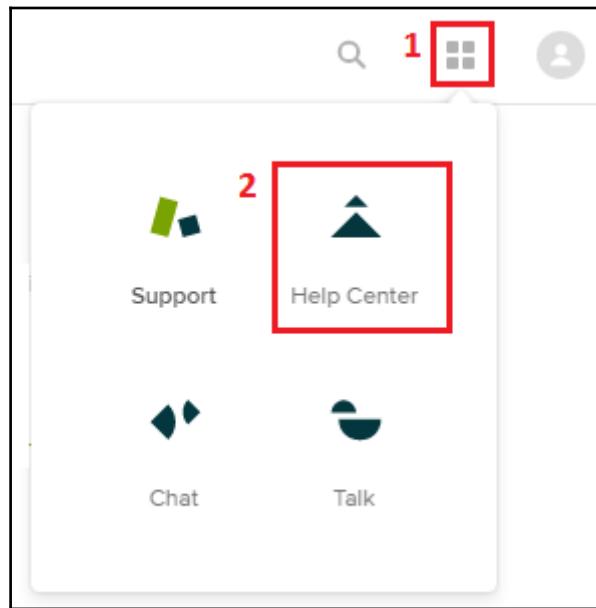
The Help Center is a platform for end-users to view predefined content in order to find the information or solution they are looking for. They can achieve this by searching the Help Centre's articles provided by us. If the end-user cannot find the right solution to their problem, they receive the option to fill out a support form in order to create a ticket.

The Help Centre has a fixed structure, allowing us to sort our articles into categories and sections. Each article needs to belong to a category and section. For example, we might have the **Software**, filled with sections, each representing another software made by ExampleComp. Within that section, we will find the articles related to that specific product, as shown here:

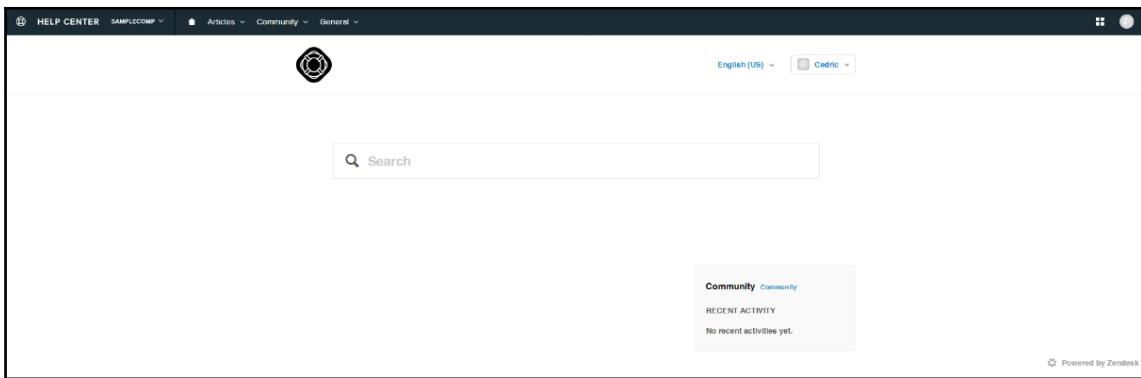


So where can we create those **Categories**, **Sections**, and **Articles**?

In order to navigate to our Help Center, simply click on the Zendesk products icon in the top bar and then click on the **Help Center** icon:



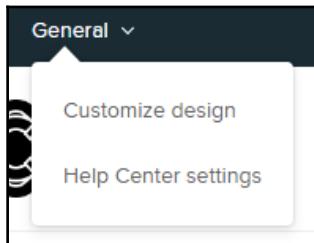
You will be presented with something like this:



It is not the best looking Help Center yet, but we can ask for help from our developers once more. While we can focus on adding the content, they can access the HTML, CSS, and JavaScript code in order to adapt the look to our brand.

By clicking on the **General** tab, you can bring up the following links:

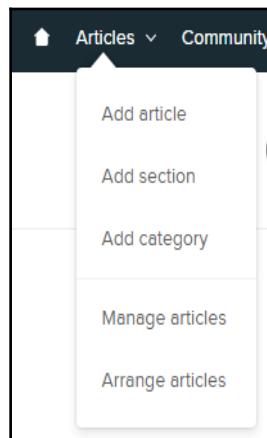
- **Customize design**
- **Help Center settings**



While our developers can, via clicking on **Customize design**, access the tools needed in order to adapt the look of the Help Center, we can take care of the overall settings by accessing the **Help Center settings**.

By clicking on the **Articles** tab, we can bring up the options we have been looking for:

- **Add article**
- **Add section**
- **Add category**
- **Manage articles**
- **Arrange articles**



Adding content is a straightforward process. So let's have a look at what an article would look like:

How to use SuperSoft

[Follow](#)

Lorum ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorum ipsum dolor sit amet. Lorum ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorum ipsum dolor sit amet.

Was this article helpful?   0 out of 0 found this helpful

Have more questions? [Submit a request](#)

End-users can read through the article first. If the content does not fit their support needs, they may click on **Submit a request** in order to create a ticket. Again, the whole look is customizable. You may, for instance, want to hide the social media sharing options.

You may also want to give users the option to submit a ticket without opening an article first by placing the button in the top navigation bar.

So what happens when we click on **Submit a request**?

The support form

Let's have a quick look:

Submit a request

Subject*

Description*

Please enter the details of your request. A member of our support staff will respond as soon as possible.

Attachments

 [Add file](#) or drop files here

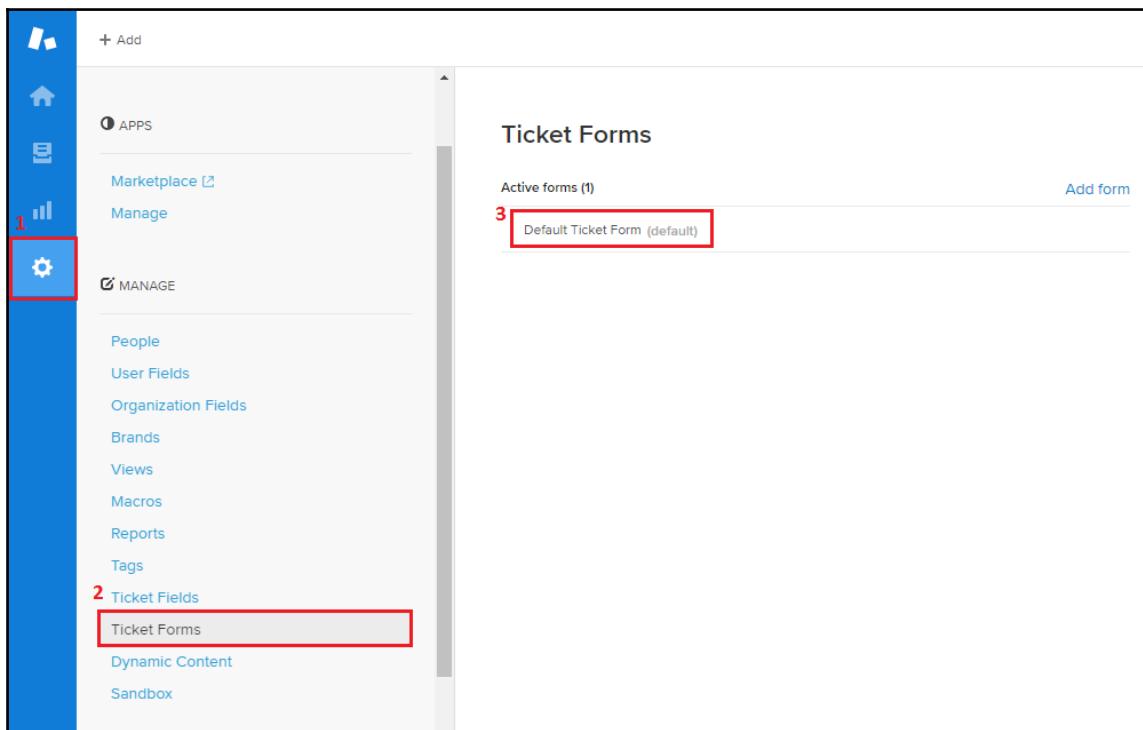
Submit

This support form shows the most basic choices for the end-user. We did, however, set up some custom fields that we did make available for our end-users.

Where are they?

Ticket Forms can be created and edited within Zendesk. So let's navigate to the corresponding page:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Ticket Forms** located under **MANAGE** within the admin menu.
3. Click on the **default** ticket form in the list to edit it:



We will be presented with a simple drag and drop system, allowing us to add our custom fields to our ticket form.

Simply drag your custom ticket fields to the form fields and click on **Save form**.

You may review the result within the Help Center.

Summary

In this chapter, you learned about the different channels provided by Zendesk. You learned about their individual purpose and how to set them up properly according to our roadmap.

I do encourage you to review the other channels as well. Rest assured, we covered enough channels necessary to understand how to set up other channels such as Voice and Chat as well. While we do not need them in our scenario, they might be interesting enough for your own project later on.

In our next chapter, you will learn about business rules and how to set up our ticket escalation processes. While focusing on our example scenario again, we will also make sure to cover every aspect of business rules. Why? Because they are awesome! Also, you will likely have to deal with them a lot.

5

Customizing Business Rules and Ticket Escalation

If I was asked to identify the most important part of any helpdesk setup, surely my answer would be “Business Rules”.

Although, I might be biased. Having a huge inclination for anything related to programming could definitely be a reason why automating processes sounds appealing to me.

Either way, the sheer amount of benefits should speak for itself. Zendesk's business rules allow us to set rules by which tickets are automatically altered. This means that we can set up conditions for automatic responses, push tickets into specific views depending on their content, and implement a bunch of other timesaving ideas we may have.

Listing all the possibilities would surely go beyond the scope of this book. So let's focus on learning how to use Business Rules by example.

In this chapter, you will look at the two different types of Business Rules—triggers and automations. You will look at all their conditions and possible actions and learn how to use them to optimize our workflows. Furthermore, we will set up our own Service Level Agreements.

This chapter covers the following topics:

- What is the difference between a trigger and an automation?
- Default triggers and automations in Zendesk.
- Available conditions and actions for both triggers and automations.
- Creating our own triggers and automations.
- Setting up Service Level Agreements.

- Managing ticket escalation with triggers and automations.
- Creating ticket views.

What are Business Rules?

We already have an idea about what Business Rules are. They are meant to automate certain processes and consist of conditions and actions. Still, it might be hard to wrap your head around the idea of Business Rules, especially if this is the first time you're dealing with such a concept.

In order to understand the idea behind Business Rules, let's look at a non Zendesk-related example.

Imagine being in charge of a food factory producing French fries. Every day, the factory receives three tons of potatoes. Currently, you have a few workers designated to handpick potatoes off the production line that do not make the cut due to their size.

This process can be automated by creating a machine taking care of this process. A simplified explanation would look something like this—if the potato is too small, then proceed by removing it from the production line.

This statement can be separated into two parts – the condition and the action.

Luckily, when it comes to Zendesk, we do not have to build the machine. We simply create the rules and Zendesk takes care of the rest.

So how does the concept translate to Zendesk?

Let's look at another example. This time we think in Zendesk terms.

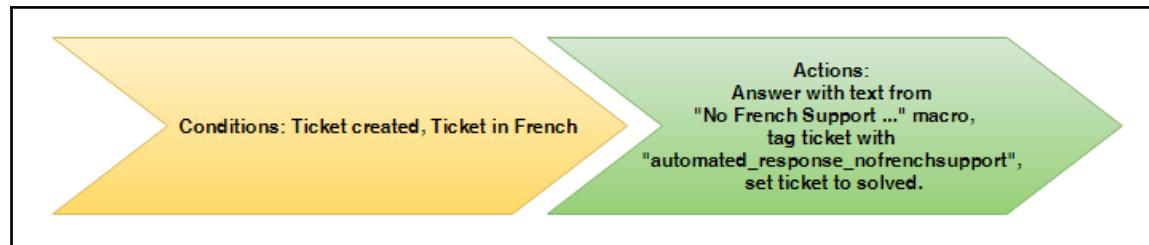
Imagine receiving about 1,000 tickets a day. About 5% of these tickets are in French, even though you do not offer support in French. Currently, your agents have to reply to these tickets manually using the designated macro called “No French Support – Please write in English”.

This process could be automated using Zendesk's Business Rules. Let's start by explaining our rule in simple English.

If we receive a new ticket with a French description, send the macro text “No French Support – Please write in English” and set the ticket to solved.

You might want to tag the ticket as well in order to keep track of this automation. A tag such as `automated_response_nofrenchsupport` can be used in our analytics reports later on.

Again, our statement can be separated into two parts—conditions and actions:



Now that you have a better understanding of the concept, let's look at the different types of Zendesk's Business Rules.

Triggers versus automations

Both triggers and automations consist of conditions and actions. The difference lies in the type of event that has to occur in order for the actions to be executed.

Every time a ticket is updated or created, Zendesk will run through all the triggers, check if the right conditions are met, and move straight to the actions.

This is not so when it comes to automations. An automation's conditions always include an extra factor, which is time. An automation would check for these specific conditions four hours after the ticket has been updated.

So when do we use an automation over a trigger?

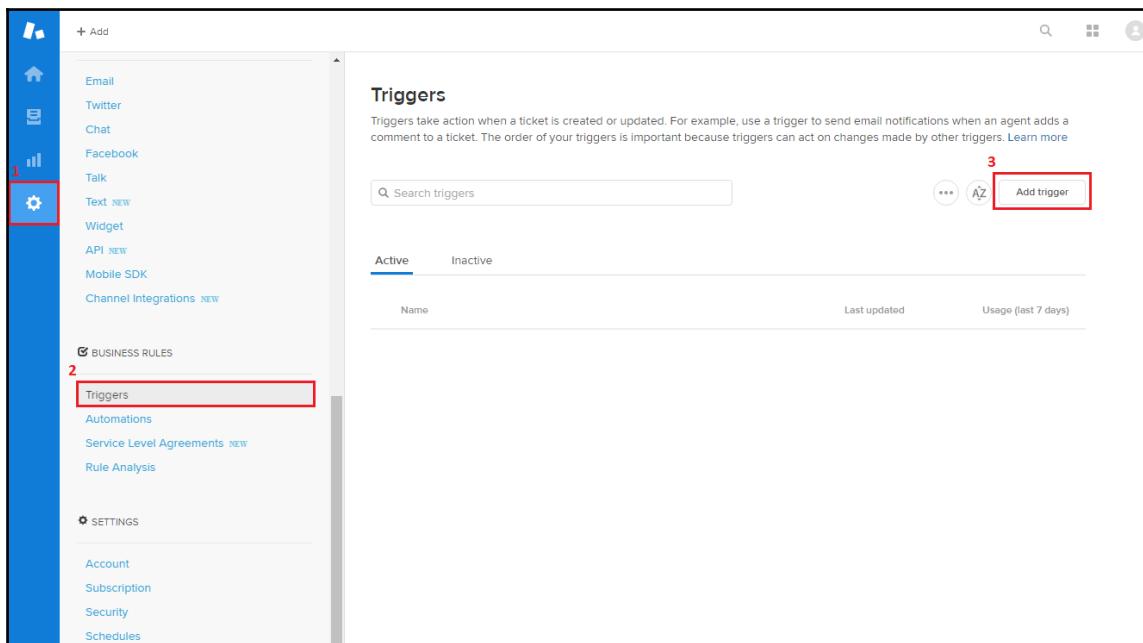
For now, remember that if you want to automate a process and include time as a condition, you will need to make use of an automation. If you want to automate a process that reacts to a certain condition straightaway, you will go for a trigger instead.

Next, let's take a detailed look at triggers.

Triggers in detail

In order to navigate to the triggers page, follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Triggers** located under **BUSINESS RULES** within the admin menu.
3. In order to create a new trigger, click on **add trigger** on the right-hand side of the screen:



We will be presented with an empty trigger waiting to be set up. We can divide this page into the following items:

- **Trigger title**
- **Meet all of the following conditions**
- **Meet any of the following conditions**
- **Perform these actions**

Trigger title

Meet **all** of the following conditions:

-- Click to select condition. --

Add condition

Meet **any** of the following conditions:

-- Click to select condition. --

Add condition

Perform these actions:

-- Click to select action. --

Add action

Create trigger

First, we can choose the **Trigger title**. It makes sense to use some sort of statement that explains the trigger's purpose. This way we understand what all our triggers do by simply reading their titles. Consider this example: *Product feedback received -> Notify Product_Owners@ExampleComp + Tag with "feedback"*:

Trigger title

Next up, we can define our trigger's conditions. There are two types of conditions. The first section allows us to set conditions of which all must be true in order for the trigger to execute. This is why it is called the **all** condition. As soon as one of them is false, the trigger will not take action:

Meet **all** of the following conditions:

-- Click to select condition. --

Add condition

From the second type of condition, called the **any** condition, only one must be true in order for the trigger to execute. If none of the conditions are met, the trigger will not perform the specified action items:

Meet **any** of the following conditions:

-- Click to select condition. --

Add condition

A lot of triggers will combine both types of conditions in order to achieve the desired outcome.

Last but not least, we can finally select our actions. Once we are happy with our trigger, we can create it by clicking on **Create trigger**:

Perform these actions:

-- Click to select action. --

Add action

Create trigger

Before looking at all possible conditions and actions, let's look at the existing default triggers to gain a better understanding of how it works and what is possible.

Default triggers

When first signing up with Zendesk, you will find that there are a few default triggers in place. You can learn a lot by examining how they work. Most likely you will want to replace them with your own versions and adapt them to your own workflows:

- Notify requester of received request
- Notify requester of comment update
- Notify assignee of comment update
- Notify assignee of assignment
- Notify assignee of reopened ticket
- Notify group of assignment
- Notify all agents of received request

Notify requester of received request

The name gives it away. The Notify requester of received request trigger will notify the requester as soon as their request has been turned into a ticket.

Let's take a look at the trigger's conditions:

- **Ticket: Is... Created**
- **Ticket: Status Is not Solved**

Trigger title

Notify requester of received request

Meet **all** of the following conditions:

Ticket: Is...	Created	-
Ticket: Status	Is not	Solved

Add condition +



Both our conditions are of the **all** type. This means both need to be true in order for the trigger to execute.

There is no need to explain the nature of these conditions as Zendesk did a great job by making all their condition and action statements self-explanatory.

So if the ticket is created and its status is not set to solved, then what?

Good question! Let's take a look at the action items of this trigger:

- **Notifications: Email user (requester)**
 - **Email subject**
 - **Email body**

The screenshot shows a configuration interface for a Zendesk trigger. At the top, it says "Perform these actions:". Below that, there are two dropdown menus: "Notifications: Email user" and "(requester)". To the right of the second dropdown is a red "X" button. Underneath these, there are two sections: "Email subject:" containing the placeholder "[Request received] {{ticket.title}}", and "Email body:" containing the text "Your request {{ticket.id}} has been received and is being reviewed by our support staff." followed by "To add additional comments, reply to this email." and the placeholder "{{ticket.comments_formatted}}".

This trigger only has one action that needs to be performed: sending a notification e-mail to the requester.

As soon as you pick an action item, it will display the corresponding fields. In this case, the **Email subject** and **Email body** fields.

One thing we notice straightaway is the use of placeholders in both the **Email subject** and **Email body** fields.

We remember the following from [Chapter 1, Configuring Your Own Zendesk](#):

Most importantly, Zendesk allows you to create dynamic content, which can be referenced by a placeholder within your macros or business rules. A placeholder would look something like this:{{dc.example_placeholder}}

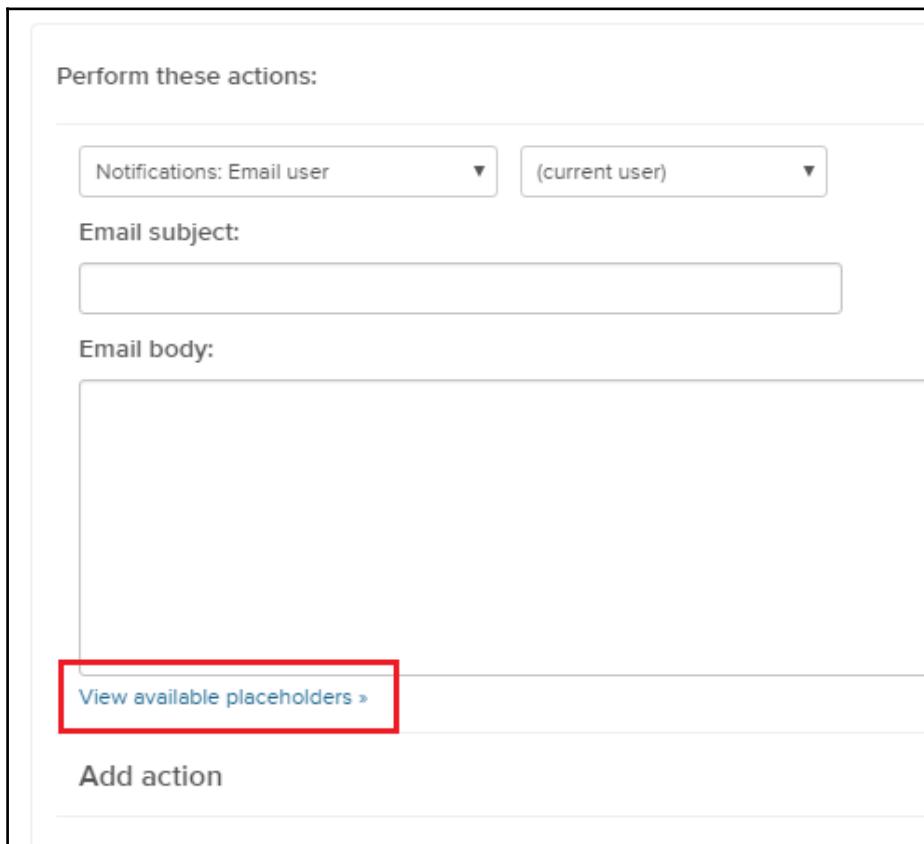
While these placeholders do look similar, these are not placeholders for dynamic content. These placeholders are part of the Zendesk placeholders that function as containers for ticket and user data.

Let's take a quick look at the placeholders used in our trigger:

- {{ticket.title}}
- {{ticket.id}}
- {{ticket.comments_formatted}}

The {{ticket.title}} placeholder will display the subject line of the ticket, {{ticket.id}} will be replaced with the unique ticket ID, and {{ticket.comments_formatted}} will show all ticket comments, with the latest comment on the top. In our case, this placeholder will basically show the description of the ticket.

These placeholders are extremely helpful when it comes to creating triggers and automations. It is, however, almost impossible to remember every available placeholder. Luckily, as soon as you add a notification action to your trigger, Zendesk will display a little link at the bottom saying **View available placeholders**:



Before we move on to the next trigger, let's look at a quote from Chapter 1, *Configuring Your Own Zendesk*:

ExampleComp is still considered a startup and cannot afford to provide customer service in more than two languages (English and German), but are planning to offer support in more languages later on.

Looking at our trigger, it is obvious that it only caters to our English customers of ExampleComp.

So what can we do?

We may feel the urge to simply add another condition like the following:



Then, if we commence by duplicating the trigger for another language, it might lead to having a huge amount of triggers.

Instead, let's make use of dynamic content. This way, the same trigger works for every language in our helpdesk environment, and it becomes a rather easy job to add another language later on.

A placeholder can be used in dynamic content as well. So, we can simply copy the text and paste it in the English version of our newly created dynamic content. Next up, we will add the German language.

Now if you ask yourself, do I need two? One for the subject line and one for the e-mail body?

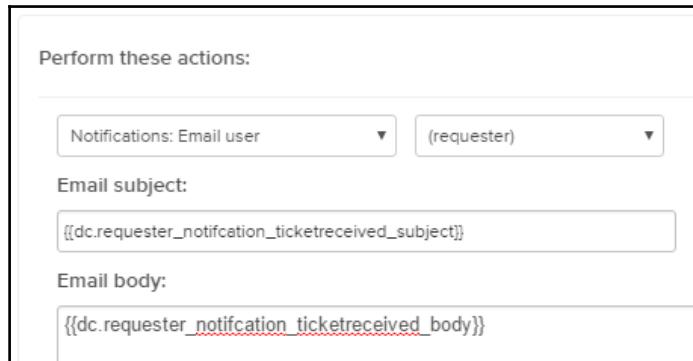
You are right. This is what the final action item will look like:

Perform these actions:

Notifications: Email user (requester)

Email subject:
{{dc.requester_notification_ticketreceived_subject}}

Email body:
{{dc.requester_notification_ticketreceived_body}}



Notify requester of comment update

Again, the name is well chosen and makes sense. This trigger will notify the requester as soon as their ticket has been updated.

Let's take a look at the trigger's conditions:

- **Ticket: Is... Updated**
- **Ticket: Comment is... Present, and requester can see the comment**
- **Ticket: Requester Is not (current user)**

Trigger title
Notify requester of comment update

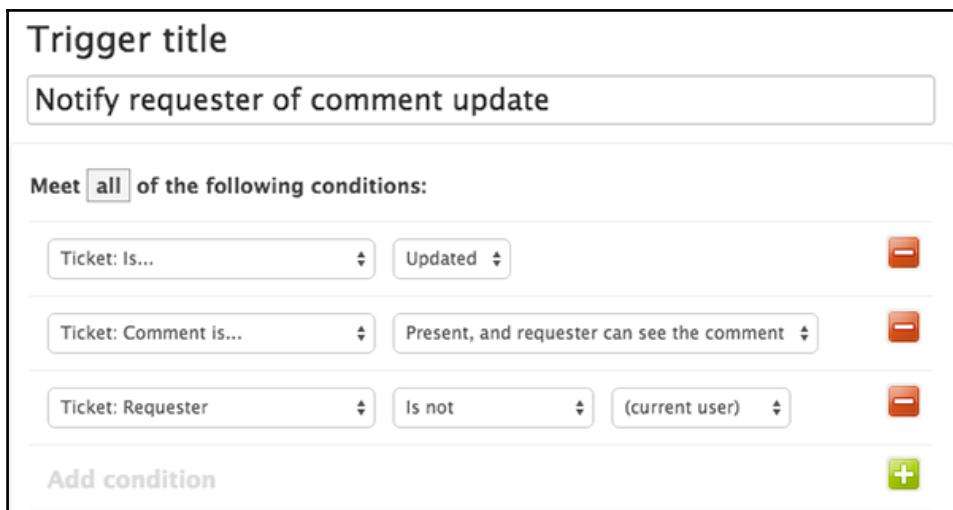
Meet **all** of the following conditions:

Ticket: Is... Updated -

Ticket: Comment is... Present, and requester can see the comment -

Ticket: Requester Is not (current user) -

Add condition +



All three conditions are of the **all** type. This means that all three of them need to be true in order for the trigger to execute.

In this case, we may want to elaborate a little. The first condition is easy to understand. As an agent adds a comment to a ticket, it counts as being updated, and this trigger is supposed to notify the requester of such updates.

However, the second condition statement might be a little confusing. It actually makes sure of two things:

1. A new comment must be present. This way, the requester is not notified by just any update.
2. The requester must also be able to see the comment.

But in what case would the requester not be able to see a comment?

Good question. There are two possible reasons for this:

1. The comment was set to private and is only meant for agents.
2. The comment was set to public, but the requester is also an agent and that specific agent does not have the necessary privilege to see such type of comments.

The last condition (The requester is not the current user) takes care of the specific case, where the requester themselves made the update to the ticket. Surely, they do not need to be notified about their own update.

So if the ticket has been updated, the comment is in fact visible to the requester, and if the current user who made the update is not the initial requester, then what?

To answer this question, let's take a look at the action items of this trigger:

- **Notifications: Email user (requester)**
 - **Email subject**
 - **Email body**

Perform these actions:

Notifications: Email user (requester) -

Email subject:
{{{ticket.account}}} Re: {{{ticket.title}}}

Email body:
Your request {{{ticket.id}}} has been updated. To add additional comments, reply to this email.
{{{ticket.comments_formatted}}}

[View available placeholders »](#)

We already know this **action** from the previous trigger. We may want to update this trigger as well by making use of dynamic content.

Notify assignee of comment update

This trigger will notify the agent as soon as the requester or another agent adds a new comment.

Let's take a look at the trigger's conditions:

- **Ticket: Comment Is... Present (public or private)**
- **Ticket: Assignee Is not (current user)**
- **Ticket: Assignee Is not (requester)**
- **Ticket: Assignee Not changed**

- **Ticket: Status Not changed from Solved**

Trigger title

Notify assignee of comment update

Meet **all** of the following conditions:

Ticket: Comment is...	Present (public or private)	-	
Ticket: Assignee	Is not	(current user)	-
Ticket: Assignee	Is not	(requester)	-
Ticket: Assignee	Not changed	-	
Ticket: Status	Not changed from	Solved	-

Add condition +

The screenshot shows a configuration interface for a trigger titled "Notify assignee of comment update". It specifies that all five listed conditions must be met for the trigger to fire. The conditions are: 1) "Ticket: Comment is..." set to "Present (public or private)" with a minus sign icon. 2) "Ticket: Assignee" set to "Is not" "(current user)" with a minus sign icon. 3) "Ticket: Assignee" set to "Is not" "(requester)" with a minus sign icon. 4) "Ticket: Assignee" set to "Not changed" with a minus sign icon. 5) "Ticket: Status" set to "Not changed from" "Solved" with a minus sign icon. A green plus sign icon is available to add more conditions.

These conditions might need some commenting in order to make sense of the combination. Let's go through them one by one.

So the comment needs to be **Present(public or private)**, which means that if there is a new comment present, be it public (from the requester) or private (from another agent), this condition is true.

The assignee should not be the **current user**, meaning, we do not need to notify the agent if they are also the one making the update to the ticket.

The ticket assignee should not be the **requester**. If the ticket assignee is the requester, there is no need to notify any of them.

The next condition basically says that the assignee needs to remain the same in order for this trigger to execute. Why? There is another trigger called *Notify assignee of assignment*, which will be executed instead. There is of course no need for two triggers to do the same work.

The last condition, saying that the ticket's status **Not changed from Solved**, has been added because of a similar reason: another trigger called *Notify assignee of reopened ticket* would already execute and let the assignee know.

So if the ticket has a new public or private comment, the assignee is not the same as the current user, is also not the same person as the requester, and has not been changed plus the status has not been changed from the status solved, then what?

To answer this question, let's take a look at the action items of this trigger:

- **Notifications: Email user (assignee)**
 - **Email subject**
 - **Email body**

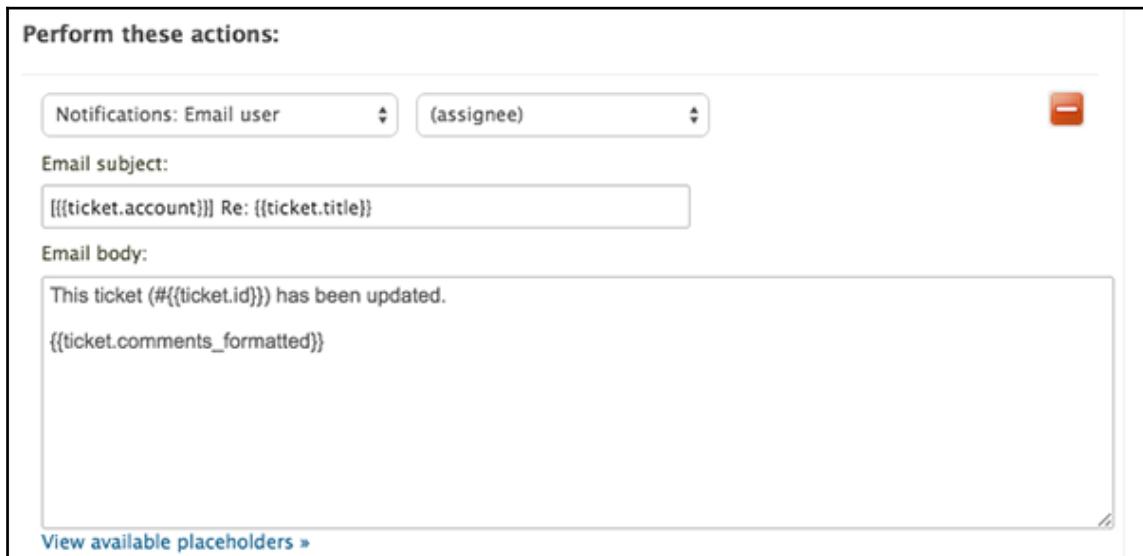
Perform these actions:

Notifications: Email user (assignee) -

Email subject:
{{{ticket.account}}} Re: {{{ticket.title}}}

Email body:
This ticket (#{{{ticket.id}}}) has been updated.
{{{ticket.comments_formatted}}}

[View available placeholders »](#)



The action is almost the same as for our previous triggers. The only difference lies in the target for our notification. Since this trigger was created in order to let the assignee know of any new comments, the assignee will also receive the e-mail notification.

In this case, we do not have to create any dynamic content as long as all of our agents speak English.

Notify assignee of assignment

This trigger is meant to notify agents as soon as a ticket has been assigned to them.

Let's take a look at the trigger's conditions:

- **Ticket: Assignee Changed**
- **Ticket: Is not (current user)**

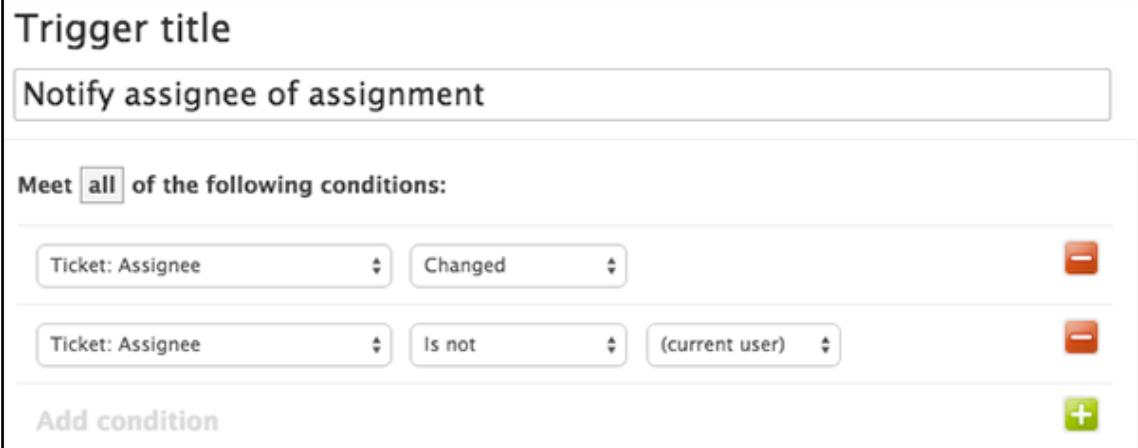
Trigger title

Notify assignee of assignment

Meet **all** of the following conditions:

Ticket: Assignee	Changed	-
Ticket: Assignee	Is not	(current user)

Add condition +



As we have gained some experience with triggers already, we can explain the conditions in just one sentence.

If the assignee changed and the new assignee is also not the one who made this change, we can move straight to the action:

- **Notifications: Email user (assignee)**
 - **Email subject**
 - **Email body**

Perform these actions:

Notifications: Email user (assignee) -

Email subject:
{{{ticket.account}}} Assignment: {{{ticket.title}}}

Email body:
You have been assigned to this ticket (#{{{ticket.id}}}).
{{{ticket.comments_formatted}}}

[View available placeholders »](#)

A screenshot of a software interface for configuring business rules or ticket escalation. It shows a section titled 'Perform these actions:' with a dropdown menu set to 'Notifications: Email user' and the recipient set to '(assignee)'. Below this are fields for 'Email subject' containing the placeholder `{{{ticket.account}}} Assignment: {{{ticket.title}}}` and 'Email body' containing the message 'You have been assigned to this ticket (#{{{ticket.id}}}). {{{ticket.comments_formatted}}}'.

This action will notify the assignee about the ticket assignment.

Notify assignee of reopened ticket

This trigger is meant to notify agents as soon as a ticket is reopened.

Let's take a look at the trigger's conditions:

- **Ticket: Assignee Is not (current user)**
- **Ticket: Status Is not Changed from Solved**
- **Ticket: Status Is not Closed**

Trigger title

Notify assignee of reopened ticket

Meet **all** of the following conditions:

Ticket: Assignee	Is not	(current user)	
Ticket: Status	Changed from	Solved	
Ticket: Status	Is not	Closed	
Add condition			

So, if the assignee did not open the ticket themselves, and the status changed from **Solved** to another status, which is not **Closed**, then this trigger will execute and perform the following actions:

- **Notifications: Email user (assignee)**
 - **Email subject**
 - **Email body**

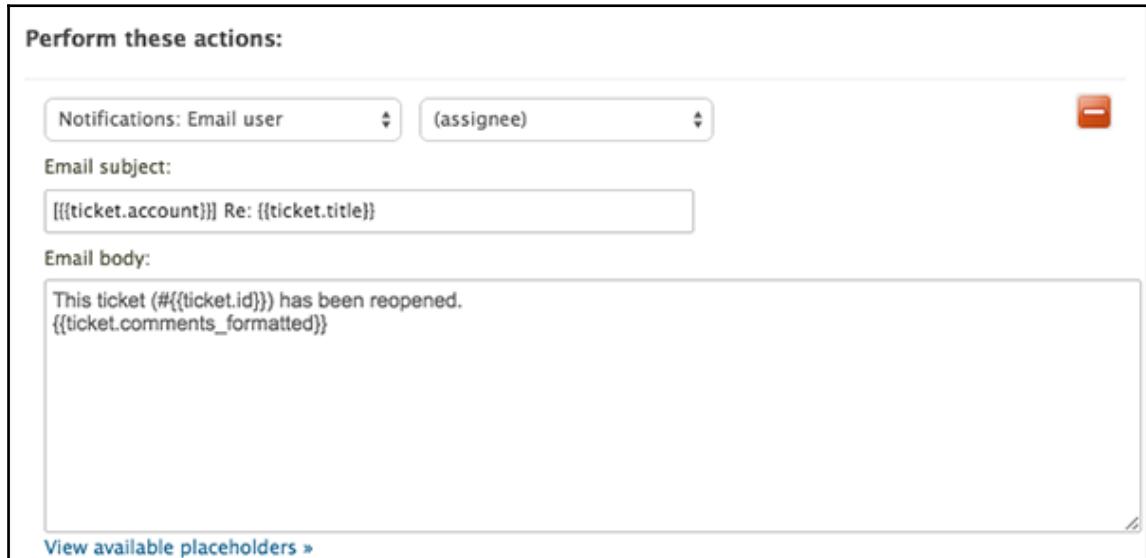
Perform these actions:

Notifications: Email user (assignee) -

Email subject:
{{{ticket.account}}} Re: {{{ticket.title}}}

Email body:
This ticket (#{{{ticket.id}}}) has been reopened.
{{{ticket.comments_formatted}}}

[View available placeholders »](#)

A screenshot of a software interface for configuring a ticket trigger. The title 'Perform these actions:' is at the top. Below it, there's a dropdown menu for 'Notifications: Email user' set to '(assignee)' with a minus sign icon to its right. Under 'Email subject', there's a text input field containing the placeholder '{{{ticket.account}}} Re: {{{ticket.title}}}'. Under 'Email body', there's a larger text area containing the placeholder 'This ticket (#{{{ticket.id}}}) has been reopened.' followed by '{{{ticket.comments_formatted}}}' and a 'View available placeholders »' link.

Notify group of assignment

This trigger is meant to notify a group of users as soon as a ticket has been assigned to their group.

This is our first trigger with both **all** and **any** conditions. Let's take a look at the trigger's **all** conditions first:

- **Ticket: Group Is not –**
- **Ticket: Assignee Is –**

Trigger title

Notify group of assignment

Meet **all** of the following conditions:

Ticket: Group	Is not	-	
Ticket: Assignee	Is	-	
Add condition			

So, as long as the ticket is definitely set to a group, not to – as in no group at all, and the ticket has not been assigned to an agent, we can check our **any** conditions next:

- **Ticket: Group Changed**
- **Ticket: Assignee Changed**

Meet **any** of the following conditions:

Ticket: Group	Changed	
Ticket: Assignee	Changed	
Add condition		

So, as long as all the **all** conditions are met and either the group changed or the assignee changed, this trigger will fire.

But why do we need to use any conditions here?

Good question. We might feel like moving the condition statement regarding the changed group up to the **all** conditions and remove the weird **assignee changed** condition altogether. But let's think about this for a moment.

What if a ticket's group has been changed, and due to the notification, an agent assigns the ticket to themselves, just to realize later that it was a mistake? The agent would remove themselves as the assignee, but no one would feel obligated anymore. This is why changing the assignee will then execute the trigger again, the group will be notified and someone else could take care of the ticket.

The action is similar to the other triggers:

- **Notifications: Email group (assigned group)**
 - **Email subject**
 - **Email body**

Perform these actions:

Notifications: Email group (assigned group) -

Email subject:
{{{ticket.account}}} "{{{ticket.group.name}}}" assignment: {{{ticket.title}}}

Email body:
This ticket (#{{{ticket.id}}}) has been assigned to group '{{{ticket.group.name}}}', of which you are a member.
{{{ticket.comments_formatted}}}

[View available placeholders »](#)

Instead of sending a notification to the assignee or requester, we are sending it to the group instead.

Notify all agents of received request

This trigger will notify all agents as soon as a new ticket has been created.

Let's take a look at the trigger's conditions:

- **Ticket: Is... Created**

Trigger title

Notify all agents of received request

Meet **all** of the following conditions:

Ticket: Is... **Created** **-**

Add condition **+**

The screenshot shows a trigger configuration window. At the top, it says 'Trigger title' and 'Notify all agents of received request'. Below that, it says 'Meet all of the following conditions:' followed by a list of conditions. The first condition is 'Ticket: Is... Created'. There are buttons to add or remove conditions. At the bottom, there is a button to 'Add condition' and a '+' sign.

No need to comment. Instead, let's take a look at the action of our trigger:

- **Notifications: Email user (all non-restricted agents)**
 - **Email subject**
 - **Email body**

Perform these actions:

Notifications: Email user (all non-restricted agents) -

Email subject:
{{{ticket.account}}} {{{ticket.title}}}

Email body:
A ticket (#{{{ticket.id}}}) by {{{ticket.requester.name}}} has been received. It is unassigned.
{{{ticket.comments_formatted}}}

[View available placeholders »](#)

Instead of sending the notification to the assignee, requester, or a group, we are sending it to **all non-restricted agents**. This means that we are sending it to all agents, whose role allows them to access every ticket.

Default trigger conclusion

Looking back at all the default triggers, we might ask ourselves:

Do we really want to send an e-mail to every agent every time a ticket has been created?

Do we need all the notifications to agents in the first place?

Would that not lead to cluttering up their e-mail tools?

The answer, however, heavily depends on our circumstances. If we receive 1000 tickets a day, some of these triggers are definitely not right for us.

It also depends on the way we handle our ticket views. If agents can review all their unsolved tickets within one dedicated view and do so on a regular basis, there is no need for a notification.

In our ExampleComp scenario, we will remove the following triggers:

- Notify assignee of comment update
- Notify assignee of assignment
- Notify assignee of reopened ticket
- Notify group of assignment
- Notify all agents of received request



Trigger Tip: When it comes to triggers, we do not always have to add the **Ticket: Is... Updated** condition. Zendesk always runs through each trigger to check their conditions as soon as a ticket has been updated anyway. The only real reason for using this conditional statement is if you specifically need to clarify that this trigger should never run, when a ticket has been created.

Conditions

After having gained a better understanding about triggers by reviewing Zendesk's default setup, you should have a closer look at the most important condition statements. Note that you may not fully understand all of the listed conditions straightaway, though each of their possible use case scenarios and advantages will become apparent in the given context later on.

Ticket-related conditions

First up, we have ticket-related conditions. They allow us to check on the values of default ticket fields or information directly connected to the ticket. Let's go through the most used conditions. While there are more conditions than listed here, you will quickly grasp how to use them by understanding the most common ones first.

Ticket: Status

This condition allows us to check for the status of the ticket:

We can use the following expressions to do so:

- Is
- Is not
- Less than
- Greater than
- Changed
- Changed to
- Changed from
- Not changed
- Not changed to
- Not changed from

The status can be checked for the following values:

- New
- Open
- Pending
- On-hold
- Solved
- Closed

Ticket: Type

This condition allows us to check the ticket type:

We can use the following expressions:

- Is
- Is not
- Changed
- Changed to
- Changed from
- Not changed
- Not changed to
- Not changed from

The ticket type can be any of the following:

- Question
- Incident
- Problem
- Task

Ticket: Priority

This condition allows us to check for the priority of the ticket:



We can make use of the following expressions:

- Is
- Is not
- Less than
- Greater than
- Changed
- Changed to
- Changed from
- Not changed
- Not changed to
- Not changed from

The priority can be checked for the following values:

- **Low**
- **Normal**
- **High**
- **Urgent**

Ticket: Group

This condition allows us to check for the group of the ticket:



We can use the following expressions:

- **Is**
- **Is not**
- **Changed**
- **Changed to**
- **Changed from**
- **Not changed**
- **Not changed to**
- **Not changed from**

The available values to check for are:

- – as in “no group”
- All custom groups set up within Zendesk

Ticket: Assignee

This condition allows us to check the assignee of the given ticket:



The following expressions are available:

- Is
- Is not
- Changed
- Changed to
- Changed from
- Not changed
- Not changed to
- Not changed from

The ticket type can be any of the following:

- – as in “no assignee”
- (current user)
- (requester)
- All agents registered within this Zendesk environment

Ticket: Requester

This condition allows us to check the requester of the given ticket:



The following expressions are available:

- Is
- Is not
- Changed
- Changed to
- Changed from
- Not changed
- Not changed to
- Not changed from

The ticket type can be any of the following:

- – as in “no requester”
- **(current user)**
- **(assignee)**
- All agents registered within this Zendesk environment

Ticket: Organization

This condition allows us to check the organization associated with the ticket in question:

A screenshot of a Zendesk search interface. It shows a search bar with the text "Ticket: Organization" followed by a dropdown arrow. To its right is another dropdown menu with the text "Is" and a dropdown arrow. Further to the right is a third dropdown menu with the text "-" and a dropdown arrow.

The following expressions are available:

- **Is**
- **Is not**
- **Changed**
- **Changed to**
- **Changed from**
- **Not changed**
- **Not changed to**
- **Not changed from**

The ticket type can be any of the following:

- – as in “no organization”
- All organizations set up by us

Ticket: Tags

This condition allows us to check for tags in our ticket:

A screenshot of a Zendesk search interface. It shows a search bar with the text "Ticket: Tags" followed by a dropdown arrow. To its right is another dropdown menu with the text "Contains at least one of the following" and a dropdown arrow. Further to the right is a large empty text input field.

The following two expressions are available:

- Contains at least one of the following
- Contains none of the following

Values to check for have to be entered manually.

Ticket: Channel

The **Ticket: Channel** condition enables us to check for the channel used to create the ticket in question:



We can choose from the following expressions:

- Is
- Is not

We can check for all the available channels:

- Web form
- E-mail
- Chat
- Twitter
- Twitter DM
- Voicemail
- Phone call (incoming)
- Phone call (outgoing)
- CTI voicemail
- CTI phone call (incoming)
- CTI phone call (outgoing)
- SMS
- Get Satisfaction
- Feedback Tab
- Web Widget
- Mobile SDK

- **Mobile**
- **Help Center post**
- **Web Service (API)**
- **Automation**
- **Forum topic**
- **Closed ticket**
- **Ticket sharing**
- **Facebook post**
- **Facebook Private Message**
- **Satisfaction Prediction**

Ticket: Update via

This condition allows us to check what channel has been used in order to update the ticket:

The screenshot shows a search bar with the query "Ticket: Update via". To the right of the search bar are two dropdown menus: one labeled "Is" and another labeled "Web form".

Expressions and values are equal to the **Ticket: Channel** condition.

Ticket: Received at

This condition gives us the option to check what e-mail address has been used in order to create the ticket:

The screenshot shows a search bar with the query "Ticket: Received at". To the right of the search bar is a single dropdown menu.

There are no extra expressions. The values consist of a list of all the e-mail addresses, set up within Zendesk.

Ticket: Satisfaction

The **Ticket: Satisfaction** condition allows us to check on the satisfaction status of the ticket:

The screenshot shows a search bar with the query "Ticket: Satisfaction". To the right of the search bar are three dropdown menus: one labeled "Is", another labeled "Unoffered", and a third one partially visible.

The following expressions can be used:

- **Is**
- **Is not**
- **Less than**
- **Greater than**
- **Changed**
- **Changed to**
- **Changed from**
- **Not changed**
- **Not changed to**
- **Not changed from**

The satisfaction can be checked for the following values:

- **Unoffered**
- **Offered**
- **Bad**
- **Bad with comment**
- **Good**
- **Good with comment**

Ticket: Is...

This condition allows checking for exactly two values. Doing so allows us to differentiate between a ticket that has just been created and a ticket that has been updated:



Therefore, the values are:

- **Created**
- **Updated**

Ticket: Comment is...

This condition allows us to check on the type of comment that was added when the ticket was updated:



The following types are possible:

- **Public**
- **Private**
- **Present (public or private)**
- **Present, and requester can see the comment**

The difference between these options might be a little hard to grasp at first. So let's look at each option once more.

A **Public** comment can be seen by everyone. A **Private** comment can only be seen by agents. A **Present (public or private)** comment tells us whether the update contained a comment in the first place. In order for the **Present, and requester can see the comment** condition to ring true, a comment must be present and also visible to the requester.

You may want to refer to the previously covered default triggers, in which their use becomes more apparent.

Ticket: Comment text...

This condition allows us to check whether specific words or strings are either present or absent in the body of the most recent comment:



We can use the following expressions:

- **Contains at least one of the following words**
- **Contains none of the following words**
- **Contains the following string**
- **Contains not the following string**

Some may wonder: what is the difference between a word and a string? While a word is usually surrounded by two spaces, a string can be found within a word. Let's look at an example to clarify—in the sentence “Download the software-update.” Zendesk would find the following words: Download, the, software-update.

If our trigger was looking for the word “software”, the condition would not be fulfilled. We should have used a string instead.



It is important to note that when a ticket is being created, these conditions will check the body as well as the subject line for the provided words and strings.

Ticket: Reopens

This condition allows us to check how often a ticket has been reopened, meaning, how often a ticket's status has been changed from solved to open or pending:

Ticket: Reopens	Is	
-----------------	----	--

The following expressions can be used:

- Is
- Less than
- Greater than

For example, you may want to escalate a ticket to a supervisor if a ticket has been reopened more than five times.

Ticket: Agent replies

This condition can be used in order to check how often agents replied via public comment:

Ticket: Agent replies	Is	
-----------------------	----	--

The following expressions can be used:

- Is
- Less than
- Greater than

Ticket: Custom Fields

This condition is truly important when it comes to customizing Zendesk, because Zendesk will automatically add this type of condition for every available custom field.

We remember from [Chapter 3, Creating Custom Fields](#) that there are two main types when it comes to customer fields—those with predefined selectable options and those without. Custom fields with predefined answers allow us to set a tag for each option.

Zendesk allows us to check these tag-based custom fields.

Requester related conditions

Next up, we have requester-related conditions.

These allow us to check on the values of default ticket fields or information directly connected to the requester. Again, let's go through the most used conditions.

Requester: Language

This condition allows us to check the language set in the requester's profile:



We can only check against languages activated in our Zendesk environment.

Requester: Role

This condition enables us to check on the requester's role, which could be any of the following:

- (agent)
- (end-user)
- (admin)

The screenshot shows a user interface for configuring business rules. A dropdown menu is open, displaying the option 'Requester: Role'. To the right of the dropdown are two buttons: one labeled 'Is' and another showing the value '(agent)'. All three elements are enclosed within a light gray rectangular border.

The following expressions can be used:

- Is
- Is not

Requester: Time zone

This condition allows us to check the requester's time zone:

The screenshot shows a user interface for configuring business rules. A dropdown menu is open, displaying the option 'Requester: Time zone'. To the right of the dropdown are two buttons: one labeled 'Is' and another showing the value '(GMT-11:00) American Samoa'. All three elements are enclosed within a light gray rectangular border.

The following expressions can be used:

- Is
- Is not

Requester: Custom Fields

This condition enables us to check the user custom fields. Again, this only works for tag-based custom fields.

Other Conditions

While most conditions are related to the ticket or the actual requester, there are some very few conditions, which do not fit into these categories. Let's take a look at these next.

Other: Current user

This condition allows us to check who was responsible for the ticket update:



The following options are available:

- (agent)
- (end user)
- A specific agent name

While we can simply choose to check whether a ticket has been updated or created by an agent or end user, we can also test if a specific agent was responsible.

The following expressions can be used:

- Is
- Is not

Organization: Custom Fields

These conditions, if available, allow us to check on custom organization fields. Just like for custom ticket and requester field conditions, the custom fields need to be tag-based in order to show up.

Actions

After going through a huge bulk of conditions, we want to know this: what actions can triggers perform? Let's review the most common actions before creating our first custom triggers.

Ticket-related actions

First up, we've got ticket-related actions. They are used in order to manipulate default and custom ticket fields when certain conditions are met.

Ticket: Status

This action allows us to change the status of the ticket in question:



The possible options are:

- New
- Open
- Pending
- On-hold
- Solved
- Closed

A ticket's status is set to **New** automatically upon creation. As soon as it gets assigned to an agent, the status changes to **Open**. When requesting additional information, the ticket is set to **Pending** until the requester's reply will change the status to **open** once more. The agent can change the status to **On-hold** while waiting for input from a third party such as the IT department. If the ticket has been addressed with the final solution, the status can be changed to **Solved**. As the requester can reopen a solved ticket again, it does make sense to set the ticket's status to **Closed** after it has been **Solved** for a couple of days. A **Closed** ticket will remain **Closed**.

Ticket: Priority

This action enables us to change the priority of a ticket:



The following values are available:

- Low
- Normal
- High
- Urgent

Changing the priority via triggers and automations allows us to create our own ticket escalation rules.

Ticket: Type

This action gives us the option to change the type of a ticket:



We can choose from the following values:

- **Question**
- **Incident**
- **Problem**
- **Task**

While these types may sound self-explanatory, there is actually a little more to them than we might expect. Here is an example of what could be achieved by utilizing the ticket type

A customer contacts our support about a bug in ExampleComp's newest software product. An agent picks up the ticket and decides to set the ticket type to **Problem**. As this problem affects many more customers, we soon receive a couple or more tickets regarding the same bug. All those tickets' types are now set to **Incident**, which consequently allows the agent to link the ticket to the initial **Problem** ticket. As the **Problem** ticket is automatically linked to a bug tracking tool, our developers have started working on a fix already. As soon as the issue has been addressed, the developer will comment the bug with the suggested solution. This comment is also displayed as a private comment in the initial **Problem** ticket. An agent can now reply to the **Problem** ticket and set it to **Solved**. This will lead to all **Incident** tickets receiving the same reply in one go.

All this workflow requires is the ticket types as well as a connection to a bug tracking tool. We will have a look at such a solution in one of the later chapters.

Ticket: Group

This action allows us to set the group of a ticket:

A screenshot of a user interface showing a dropdown menu. The menu is labeled "Ticket: Group" and contains two items: a dash (-) and another item whose name is partially visible.

The following options are available:

- - as in “no group”
- Specific group name

Ticket: Assignee

This action allows us to set the assignee of the ticket:

A screenshot of a user interface showing a dropdown menu. The menu is labeled "Ticket: Assignee" and contains several items: a dash (-), "(requester)", "Agent name", and "(current user)".

The following values can be used:

- – as in “no assignee”
- **(requester)**
- Agent name
- **(current user)**

Ticket: Satisfaction

This action only has one purpose. Marking the ticket as soon as we have sent out a satisfaction survey:

A screenshot of a user interface showing a dropdown menu. The menu is labeled "Ticket: Satisfaction" and contains one item: "offered to requester".

We will use this later on when looking at default automations.

Ticket: Set tags

This action allows us to set the ticket's tags:

Ticket: Set tags	▼	<input type="text"/>
This action will remove existing tags and replace with the tags you specify (use spaces to separate individual tags)		

It is important to note that it will replace all pre-existing tags.

Ticket: Add tags

This action allows us to add tags to a ticket:

Ticket: Add tags	▼	<input type="text"/>
This action will add the tags you specify to existing tags (use spaces to separate individual tags)		

Contrary to the setting tags, it will not replace all previous tags.

Ticket: Remove tags

This action allows us to remove specific tags:

Ticket: Remove tags	▼	<input type="text"/>
This action will remove the tags you specify from ticket if present (use spaces to separate individual tags)		

Ticket: Add CC

This action allows us to add an agent or the (**current user**) to the CC field of a ticket:

Ticket: Add CC	▼	(current user)	▼
----------------	---	----------------	---

Once added to the CC field, that user will also be notified about ticket updates.

Ticket: Custom Fields

These actions allow us to manipulate tag-based custom ticket fields.

Requester- and organization-related actions

Next up, we have requester- and organization-related actions. They are used in order to manipulate default and custom user/organization fields when certain conditions are met.

Requester: Language

This actions allows us to change the language of the requester:



Requester: Custom fields

These actions allow us to manipulate tag-based custom user fields of the requester.

Organization: Custom fields

These actions allow us to manipulate tag-based custom organization fields.

Other actions

Let's take a look at some actions that are not connected to any Zendesk fields. After all, triggers are meant for more than only changing ticket values.

Notifications: Email user

This action allows us to send a notification to a user:

The screenshot shows a configuration interface for sending an email notification. At the top left, there is a dropdown menu labeled "Notifications: Email user". To its right is another dropdown menu labeled "(requester)". Below these are two input fields: one for "Email subject" and one for "Email body", both represented by large, empty text areas. At the bottom of the interface, there is a blue link labeled "View available placeholders »".

First, we can choose who to send this notification to:

- **(current user)**
- **(requester)**
- **(assignee)**
- **(all non-restricted agents)**
- A specific agent

Next, we simply choose the **Email subject** and **the Email body** for the notification.

Notifications: Email group

This action enables us to send notifications to whole groups of users:

The screenshot shows a configuration interface for an 'Email group' notification. At the top left is a dropdown menu labeled 'Notifications: Email group'. To its right is another dropdown menu labeled '(assigned group)'. Below these are two input fields: 'Email subject:' and 'Email body:', each with a large text area for input. At the bottom of the interface is a blue link labeled 'View available placeholders »'.

We can choose to send this notification to the ticket's assigned group as well as to a specific group:

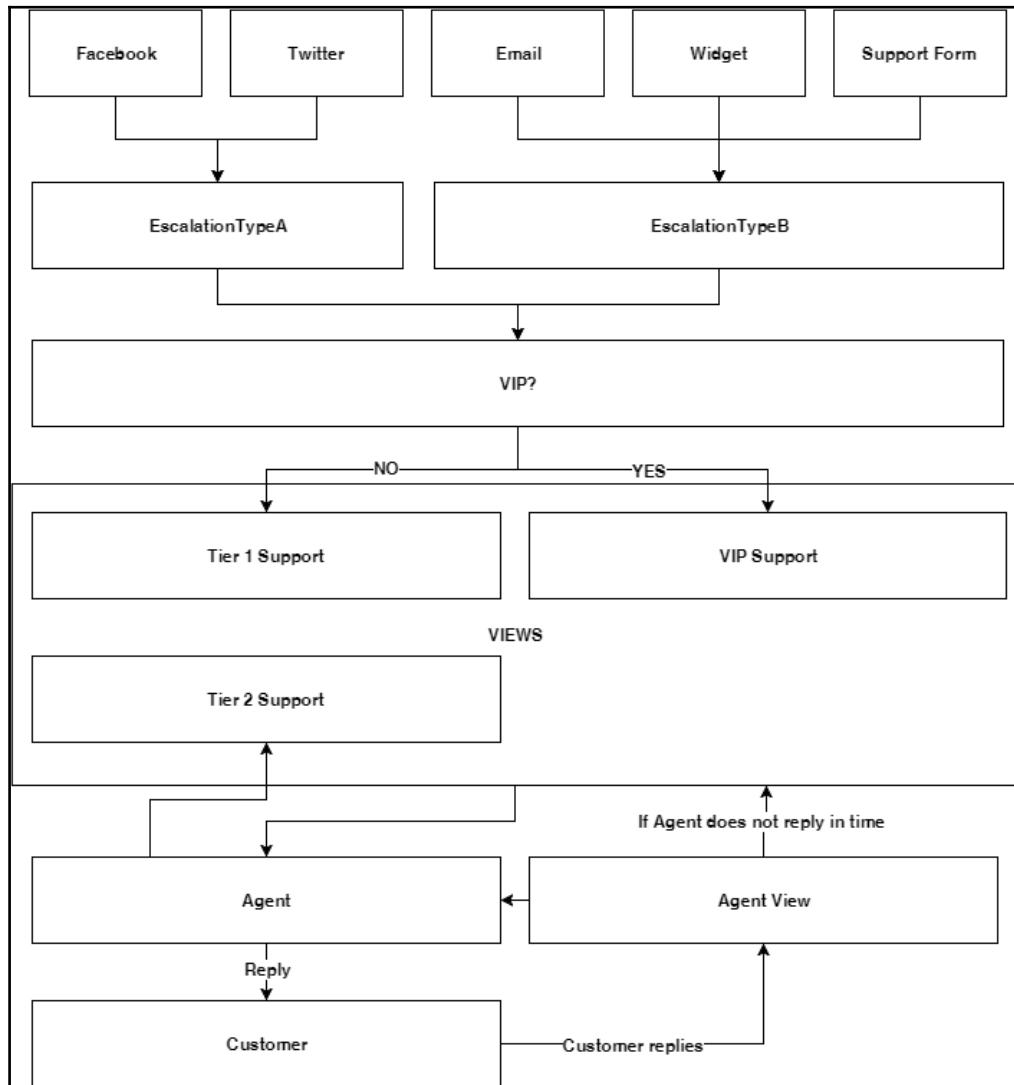
- **(assigned group)**
- A specific group

Again, we can enter an e-mail subject and body to make this action complete.

Creating custom triggers

Finally! We made it through a whole lot of information in one go. Let's put this newly gained knowledge to use by creating our own custom triggers.

Before we can trigger away, we need to review our flowchart from *Chapter 1, Configuring Your Own Zendesk* again:



By looking at our flowchart we can derive potential triggers. In our case, there are two triggers we can take care of now:

- Tagging tickets as **EscalationTypeA**
- Tagging tickets as **EscalationTypeB**

From Chapter 1, *Configuring Your Own Zendesk* we also remember the following:

... not all channels should be treated equally. Meaning that different channels will require different SLA rules. It is common practice, for instance, that support requests created via Facebook are being escalated quicker than tickets created via the e-mail channel.

We also established the following:

Knowing that group 1 and group 2 have to be handled differently when it comes to escalation rules, we can take note, that both groups have to be marked differently when the ticket is being created. Let's name each group. In this case, since we only have two different groups, we do not have to be too descriptive when naming them:

EscalationTypeA

EscalationTypeB

Reading this now, it may be a little confusing as we referred to the two different escalation types as groups, which in Zendesk terms they are not. It makes way more sense to use business rules to tag the tickets accordingly.

So what we need to do is create two different triggers that will tag tickets according to their channels. Let's get right into it:

Trigger title
SLA Rules - Channels: Facebook, Twitter -> Tag ticket with "EscalationTypeA"

Meet **all** of the following conditions:

Ticket: Is... **Created** -

Add condition +

Meet **any** of the following conditions:

Ticket: Channel **Is** **Twitter** -

Ticket: Channel **Is** **Facebook Private Message** -

Add condition +

Perform these actions:

Ticket: Add tags **EscalationTypeA** -

This action will add the tags you specify to existing tags (use spaces to separate individual tags)

Add action +

Create trigger

Our first trigger fires as soon as a ticket has been created using either the **Facebook** or the **Twitter** channel. If all the conditions are met, it will commence by tagging the ticket with **EscalationTypeA**.

Our second trigger is very similar:

Trigger title
SLA Rules - Channels: Email, Widget, Web Form -> Tag ticket with "EscalationTypeB"

Meet **all** of the following conditions:

Ticket: Is... Is Created

Add condition

Meet **any** of the following conditions:

Ticket: Channel Is Email

Ticket: Channel Is Web Widget

Ticket: Channel Is Web form

Add condition

Perform these actions:

Ticket: Add tags EscalationTypeB

This action will add the tags you specify to existing tags (use spaces to separate individual tags)

Add action

Create trigger

The screenshot shows a configuration interface for a ticket trigger. It starts with a title 'SLA Rules - Channels: Email, Widget, Web Form -> Tag ticket with "EscalationTypeB"'. Below the title, it says 'Meet all of the following conditions' and lists three conditions: 'Ticket: Is... Is Created'. There is a 'Create' button next to this section. Below that, it says 'Meet any of the following conditions' and lists three more conditions: 'Ticket: Channel Is Email', 'Ticket: Channel Is Web Widget', and 'Ticket: Channel Is Web form'. Each of these has a 'Create' button to its right. After these sections, there is a 'Perform these actions' section with a single action 'Ticket: Add tags EscalationTypeB'. A note below it says 'This action will add the tags you specify to existing tags (use spaces to separate individual tags)'. This section also has a 'Create' button. At the bottom right of the entire configuration area is a large 'Create trigger' button.

Our second trigger fires as soon as a ticket has been created using either the **Email**, **Web Widget**, or **Web form** (Support Form) channel. If all the conditions are met, it will commence by tagging the ticket with **EscalationTypeB**.

Great, but how does tagging these tickets get us anywhere?

Tagging our tickets is only the first step. It allows us to escalate tickets differently depending on their “escalation type”. Since we want to escalate tickets to different priority levels based on time, we will need to tackle the next step by creating the necessary automations.

But why do we tag the tickets using triggers? Is it not easier to let the automations decide what tickets to escalate at what speed?

Great question! We use triggers for the initial assignment of the “escalation type” for a few reasons:

- It allows us to add channels or other criteria by manipulating two triggers instead of having to go through eight or more automations later. Why? Each ticket can be escalated from **Low** to **Normal** to **High** to **Urgent**. We decided on two different types of tickets that would escalate at different speeds. In order to achieve this, we need six to eight automations.
- It allows us to create smaller automations, which look cleaner and are quicker to read.

Additional tips

Dealing with triggers will become easier over time. Here are a few tips when it comes to triggers.

Naming and organizing triggers

Naming triggers can be a challenging task. Obviously, we want to make sure that our triggers are easy to understand. Grasping a trigger's purpose simply by reading its name would be the ideal outcome.

In general, we should try to make sure that each title explains what the trigger actually does. But there is something else that we can do: we can add list dividers.

While Zendesk does not provide an option as such, we can get around that by creating fake triggers to serve as dividers in our list. For instance, we could create a trigger that checks for the following tag—`this tag will never be added to a ticket`.

We can then change the name of the trigger to something like this—“`////////// SLA Triggers ///////////////`”.

Let's look at an example:

New SLA Triggers
SLA - Group - Normal
SLA - Group - Dynamic
SLA - Group - Call (Sippgate)
SLA - Group - Payment
SLA - Group - Social Media + Impressum
SLA - Group- 3rd_level

Testing triggers

Creating triggers in a live setup can be quite challenging as well. Imagine working on a trigger that is supposed to manipulate hundreds or even thousands of tickets. We definitely want to make sure that the trigger is doing what it is supposed to do.

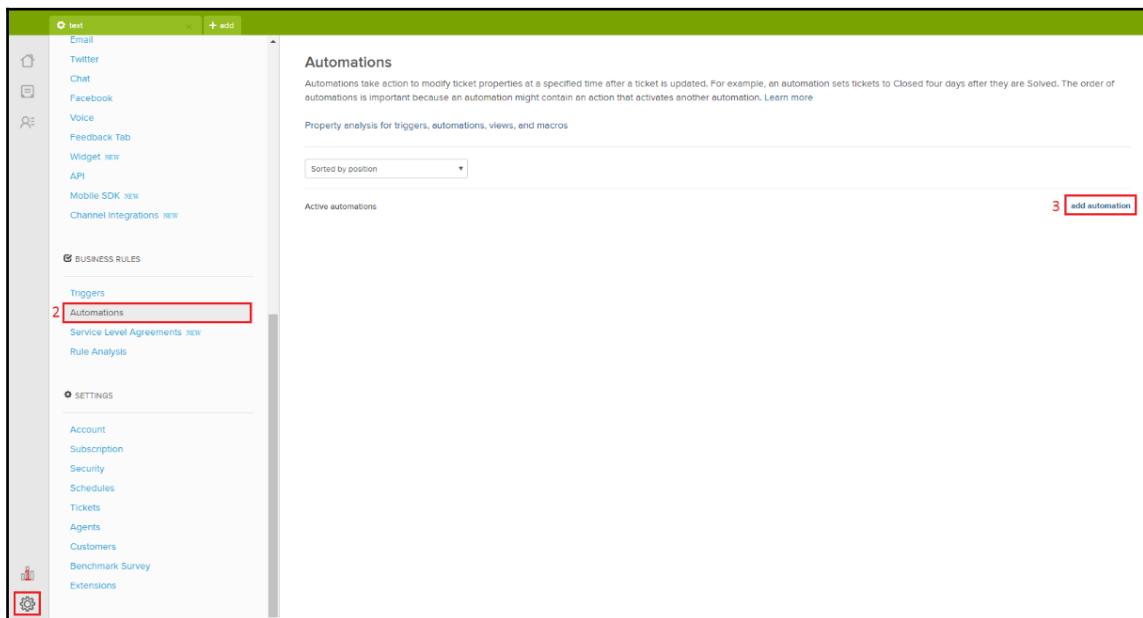
One way to test a trigger is by adding an extra condition such as this one—Ticket : Comment text contains at least one of the following strings:
"test_4231".

This will allow us to create a test ticket in order to preview our trigger first.

Automations in detail

In order to navigate to the automations page, follow these steps:

1. Click on the Admin icon (gear symbol) located at the bottom of Zendesk's sidebar.
2. Click on **Automations** located under **BUSINESS RULES** within the admin menu.
3. In order to create a new automation, click on **add automation** on the right side of the screen:



We will be presented with an empty automation waiting to be set up. We can divide this page into the following items:

- **Automation title**
- **Meet all of the following conditions**
- **Meet any of the following conditions**
- **Perform these actions**

Automation title

Meet **all** of the following conditions:

-- Click to select condition. -- ▾

Add condition - +

Meet **any** of the following conditions:

-- Click to select condition. -- ▾

Add condition - +

Preview match for the conditions above

Perform these actions:

-- Click to select action. -- ▾

Add action - +

Create automation

First we can choose the **Automation title**. Again, it makes sense to use some sort of statement that explains the automation's purpose. This way, we understand what all our automations do by simply reading through their titles. Look at this example—Ticket over 7 days old -> Notify Teamleader + Tag with "old_request".

Automation title

Next up, we can define our automation's conditions. Just like for triggers, we can add the **all** and **any** conditions:

The screenshot shows the configuration for a 'Meet all' condition. It includes two sections: 'Meet all of the following conditions:' and 'Meet any of the following conditions:'. Each section has a dropdown menu labeled 'Click to select condition.' and a red minus sign icon. Below each section is an 'Add condition' button with a green plus sign icon.

A lot of automations will combine both types of conditions in order to achieve the desired outcome. Before adding our actions, we can click on the **Preview match for the conditions above** button, which will present us with a list of tickets that match our conditions:

Preview match for the conditions above

Last but not least, we can finally select our **actions**. Once we are happy with our automation, we can create it by clicking on **Create automation**:

The screenshot shows the configuration for performing actions. It includes a section for 'Perform these actions:' with a dropdown menu labeled 'Click to select action.' and a red minus sign icon. Below this is an 'Add action' button with a green plus sign icon. In the bottom right corner is a dark blue 'Create automation' button.

Before looking at the exact differences between triggers and automations, let's look at the existing default automations to gain a better understanding of how it works and what is possible.

Default automations

The best way to learn about automations would be by examining how Zendesk's default automations work. Most likely, just like for the triggers, you will want to replace them with your own versions and adapt them to your own workflows:

- Close ticket 4 days after status is set to solved
- Request customer satisfaction rating

Close ticket 4 days after status is set to solved

This automation will set the ticket's status to closed after it has been solved for four days.

Let's take a look at the automation's conditions:

- Status Is Solved
- Hours since solved Greater than 96

Automation title

Close ticket 4 days after status is set to solved

Meet **all** of the following conditions:

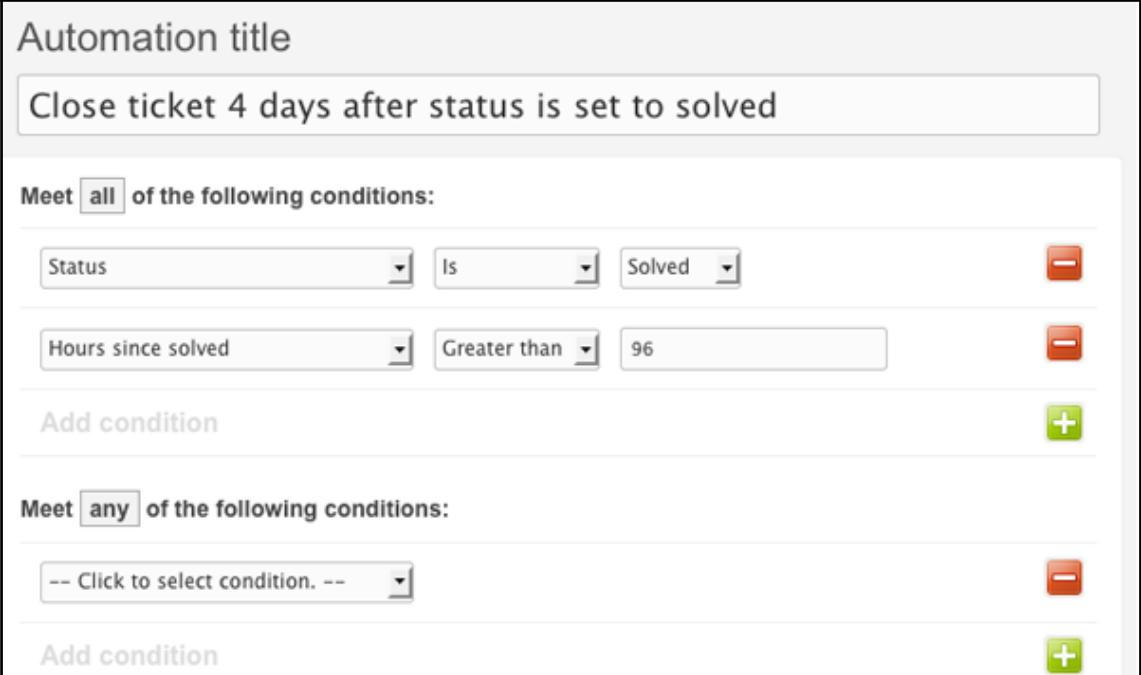
Status	Is	Solved	-
Hours since solved	Greater than	96	-

Add condition **+**

Meet **any** of the following conditions:

-- Click to select condition. --	-
----------------------------------	---

Add condition **+**



Both conditions are of the **all** type. This means that both of them need to be true in order for the automation to execute. They are pretty straightforward, so let's take a look at the actions:



This automation only has one action. As soon as the ticket has been solved for over 96 hours, Zendesk will automatically set the ticket's status to **Closed**.

Request customer satisfaction rating

This automation is designed to send a notification to the requester asking them to rate their support experience.

Let's take a look at the automation's conditions:

- Status Less than Closed
- Hours since solved (calendar) Is 24

- **Ticket Satisfaction Is Unoffered**

Automation title

Request customer satisfaction rating (System Automation)

Meet **all** of the following conditions:

Status	Less than	Closed	-
Hours since solved	(calendar) Is	24	-
Ticket Satisfaction	Is	Unoffered	-

Add condition +

Meet **any** of the following conditions:

-- Click to select condition.	-
-------------------------------	---

Add condition +

All three conditions are of the **all** type. This means that all of them need to be true in order for the automation to execute. Let's examine these conditions a little further.

The **Status** must be **Less than Closed**, meaning the ticket's status must be anything but **Closed**. The **Less than** expression makes more sense when looking at the dropdown's content:



The **Less than** expression rings true for any value before the value picked in the following dropdown.

The **Hours since solved** value has to be exactly **24**. The **(calendar) Is** option refers to the fact that every hour is counted, not just business hours, which can be set in Zendesk.

Ticket Satisfaction Is Unoffered must be true, making sure that this automation can only fire once. This will make more sense when looking at the actions:

Perform these actions:

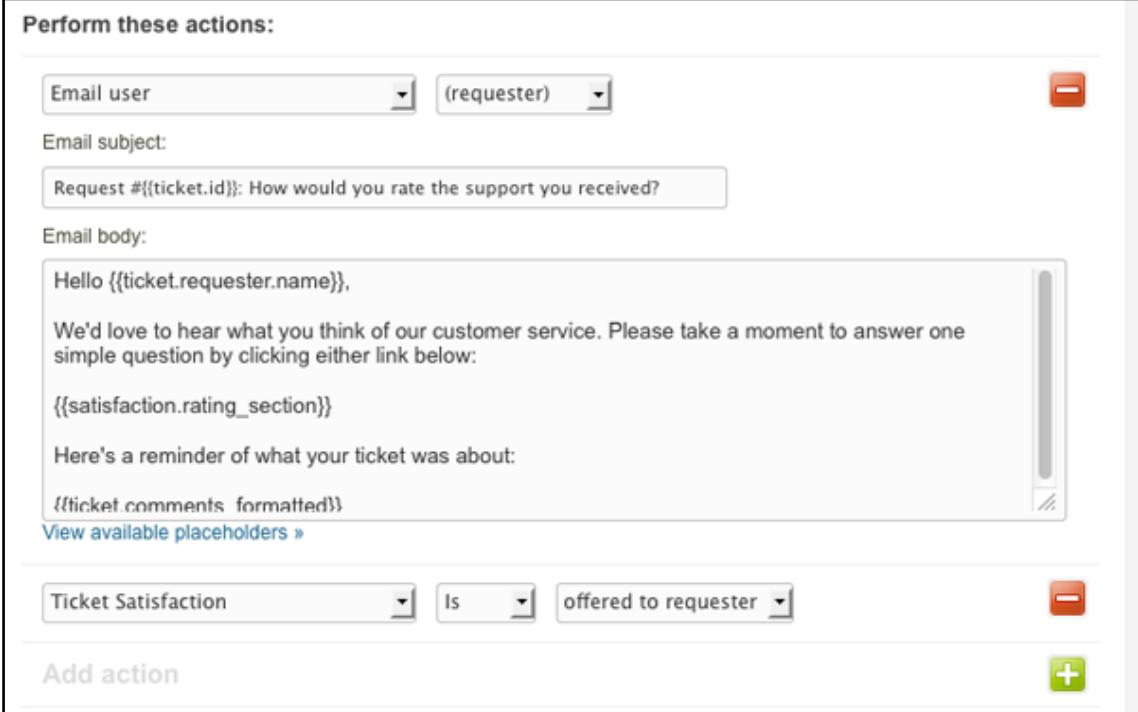
Email user (requester) -

Email subject:
Request #{{ticket.id}}: How would you rate the support you received?

Email body:
Hello {{ticket.requester.name}},
We'd love to hear what you think of our customer service. Please take a moment to answer one simple question by clicking either link below:
{{satisfaction.rating_section}}
Here's a reminder of what your ticket was about:
{{ticket.comments_formatted}}
[View available placeholders »](#)

Ticket Satisfaction Is offered to requester -

Add action +



This automation only has two actions. As soon as all conditions are met, this automation will proceed and send an e-mail to the requester.

This notification will make use of placeholders in order to present the receiver with the option to rate their customer experience.

The second action will mark the ticket by setting **Ticket Satisfaction** to **offered to requester**. As our conditions do not allow this automation to fire, whenever the ticket has been marked as such already, this will ensure that the automation only fires once per ticket.

Automation-specific conditions

As most conditions and actions are the same for triggers and automations, instead of going through all the existing conditions and actions again, let's look at those specific conditions that can only be found when working with automations:

- **Ticket: Hours since created**
- **Ticket: Hours since open**
- **Ticket: Hours since pending**
- **Ticket: Hours since on-hold**
- **Ticket: Hours since solved**
- **Ticket: Hours since closed**
- **Ticket: Hours since assigned**
- **Ticket: Hours since update**
- **Ticket: Hours since requester update**
- **Ticket: Hours since assignee update**
- **Ticket: Hours since due date**
- **Ticket: Hours until due date**
- **Ticket: Hours since last SLA breach**
- **Ticket: Hours until next SLA breach**

Meet **all** of the following conditions:

Ticket: Status	Less than	New
Ticket: Channel		
Ticket: Received at		
Ticket: Satisfaction		
Ticket: Hours since created		
Ticket: Hours since open		
M Ticket: Hours since pending		
Ticket: Hours since on-hold		
Ticket: Hours since solved		
Ticket: Hours since closed		
Ticket: Hours since assigned		
Ticket: Hours since update		
Ticket: Hours since requester update		
Ticket: Hours since assignee update		
Ticket: Hours since due date		
Ticket: Hours until due date		
Ticket: Hours since last SLA breach		
Ticket: Hours until next SLA breach		
Requester: Role		
Requester: Language		

Most of these are pretty straightforward and there is no need for further explanations. The last four, however, may need some additional information:

- **Ticket: Hours since due date**
- **Ticket: Hours until due date**
- **Ticket: Hours since last SLA breach**
- **Ticket: Hours until next SLA breach**

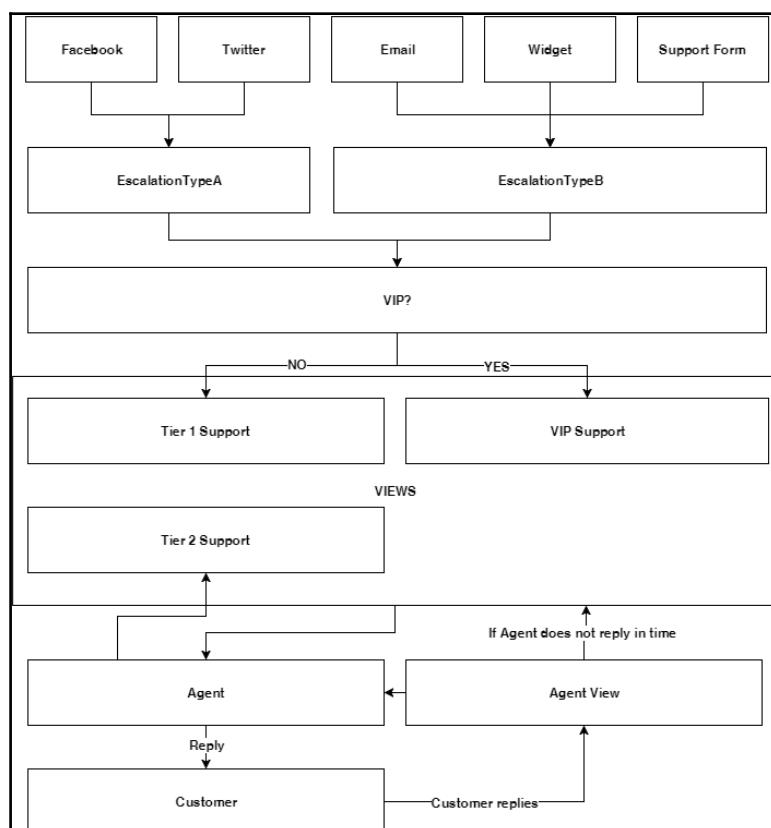
These conditions are related to Zendesk's SLA system. We remember the following from Chapter 1, *Configuring Your Own Zendesk*:

A Service Level Agreement or SLA can be seen as a contract between you as a service provider and your customers, outlining the level of service that can be expected in a certain timeframe. This feature is also helpful when using reports to review your team's performance.

So as soon as we set our own SLAs, we can make use of these conditions. It will become clearer later when looking at how SLAs work in Zendesk.

Creating custom automations

Let's move ahead and create our own automations. Again, we should take a look at our flowchart first:



By looking at our flowchart we can derive potential automations. In our case, there are quite a few automations that we can take care of now:

- If the agent does not reply in time, unassign the ticket
- Change ticket priority depending on “escalation type” and hours since created

Let's take care of our first automation. We remember the following from [Chapter 1, Configuring Your Own Zendesk](#):

But what happens when a customer replies?... The ticket could still be assigned to the same agent, who would then commence helping that customer if possible. The agent could receive an e-mail, notifying him about the customer's reply. But what if our agent stopped working already? Well, a business rule could remove the assignee (assigned agent) after a specific amount of time since the last ticket update. Doing that would force the ticket back into its initial view where another agent could then take care of the customer.

So if a ticket is assigned to an agent, the ticket is set to open (because the customer replies) and if the agent has not responded within three hours after the last update by the customer, we will remove the assignee and force the ticket back into its public view.

The automation we are looking for, would look something like this:

Automation title

Requester replies + 3 hours no agent interaction -> unassign ticket

Meet **all** of the following conditions:

Ticket: Status	Is	Open
Ticket: Assignee	Is not	-
Ticket: Hours since open	(business) Greater than	3

Add condition

Perform these actions:

Ticket: Assignee	-
------------------	---

Add action

The screenshot shows the Zendesk Automation configuration interface. It starts with a title 'Requester replies + 3 hours no agent interaction -> unassign ticket'. Below it, a condition is defined: 'Meet all of the following conditions'. Three conditions are listed: 'Ticket: Status Is Open', 'Ticket: Assignee Is not -', and 'Ticket: Hours since open (business) Greater than 3'. Below the conditions, an 'Actions' section is shown with a single action: 'Ticket: Assignee -'. The entire configuration is contained within a large rectangular frame.

In order to change a ticket's priority depending on “escalation type” and hours since its creation, we need to set up some rules beforehand.

We established two types of tickets—EscalationTypeA and EscalationTypeB.

EscalationTypeA refers to tickets coming through social media channels. We may want to reply to them as soon as possible.

Here is an example of how we can set up our automations for EscalationTypeA:

- Ticket is 1 (business hours) old -> Low Priority
- Ticket is 2 (business hours) old -> Normal Priority
- Ticket is 4 (business hours) old -> High Priority
- Ticket is 5 (business hours) old -> Urgent Priority

Here is an example of how we can set up our automations for EscalationTypeB:

- Ticket is 8 (business hours) old -> Low Priority
- Ticket is 12 (business hours) old -> Normal Priority
- Ticket is 16 (business hours) old -> High Priority
- Ticket is 24 (business hours) old -> Urgent Priority

For a setup like this, we would need eight automations in total. Here is what one of them would look like:

Automation title

SLA - EscalationTypeB - 8 business hours -> Low Priority

Meet **all** of the following conditions:

Ticket: Status ▼ Less than ▼ Solved ▼ -

Ticket: Hours since created ▼ (business) Is ▼ 8 -

Ticket: Tags ▼ Contains at least one of the following ▼ -

EscalationTypeB

Add condition +

Meet **any** of the following conditions:

-- Click to select condition. -- -

Add condition +

Preview match for the conditions above

Perform these actions:

Ticket: Priority ▼ Low ▼ -

Add action +

This automation is pretty straightforward. Using this example, we can go ahead and create the rest of them as well.



Keep in mind that this setup is just an example, showcasing the possibilities of automations. There are many ways to deal with ticket priorities and SLAs in general. You may want to create a way more complex system or change the priority depending on the keywords in your customer's request.

Custom ticket views

In order to take advantage of our ticket's automatic priority system, we will need to create views that sort tickets according to those values. Let's take a quick look at how we would set up such a view.

In order to navigate to the Views page, follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Views** located under **MANAGE** within the admin menu.
3. In order to create a new view, click on **Add view** on the right side of the screen:

The screenshot shows the Zendesk Admin interface. On the left, there is a sidebar with various icons and a list of categories under 'MANAGE'. The 'Views' item under 'Brands' is highlighted with a red box and labeled '2'. The 'Add view' button in the top right corner of the main content area is also highlighted with a red box and labeled '3'. The main content area is titled 'Views' and contains a search bar, sorting options ('Active' and 'Inactive'), and a link to 'All shared views'.

We will be presented with an empty View waiting to be set up. We can divide this page into the following items:

- **View title**
- **Meet all of the following conditions**
- **Meet any of the following conditions**
- **Formatting options**

Setting up a view is not too different from setting up a trigger or automation. First we pick our conditions, followed by our preferred formatting options.

For ExampleComp, we will need three views according to our road map:

- Tier 1 Support
- Tier 2 Support
- VIP Support

Our “Tier 1 Support” view should list tickets that are either *new* or *open*, sorted by *priority* without the `tier2` and `vip` tags.

Our second view, called “Tier 2 Support”, should show *new* and *open* tickets, which are tagged `tier2` and not tagged with `vip`, sorted again by priority.

While we did not create an option for agents to move tickets from Tier 1 support to Tier 2 support, we can keep in mind that the mechanism we come up with should simply tag the tickets in order to show up in the right view.

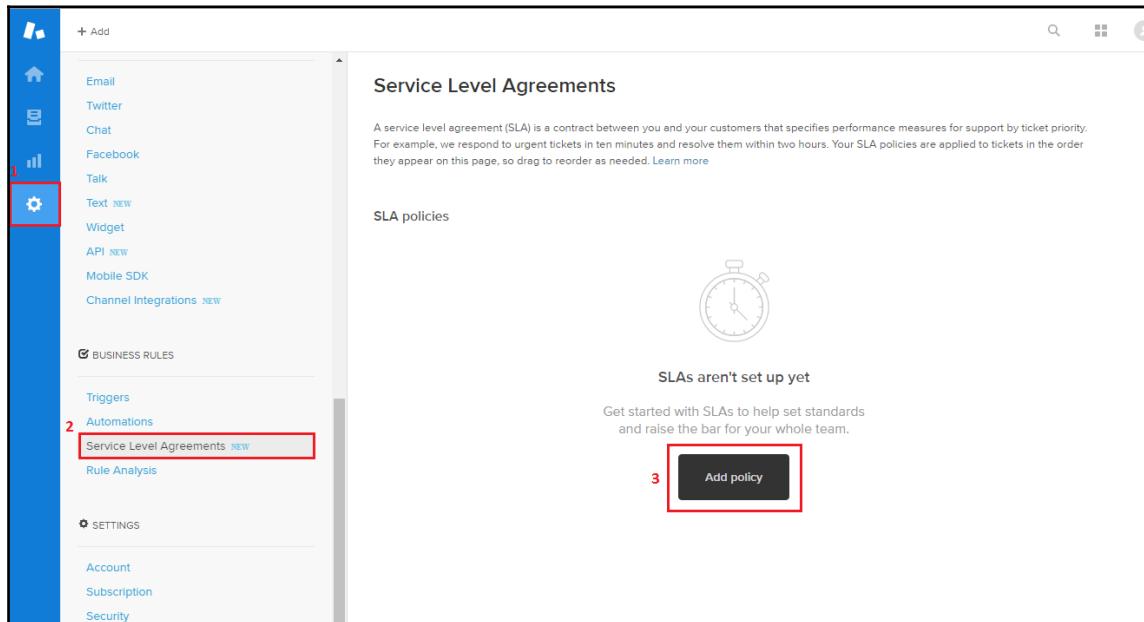
Last but not least, we can add our VIP view.

Service Level Agreements

Setting up SLAs in Zendesk is a pretty straightforward task. Similar to triggers and automations, you are asked to pick a name for your policy followed by conditions.

In order to navigate to the SLA page, follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Service Level Agreements** located under **BUSINESS RULES** within the admin menu.
3. In order to create a new policy, click on **Add policy**:



We will be presented with an empty policy waiting to be set up. We can divide this page into the following items:

- Policy Name and Description
- Meet all of the following conditions
- Meet any of the following conditions
- Targets

Since we focused on conditions quite extensively when dealing with triggers and automations, we will focus on targets instead:

Targets				
	Urgent	High	Normal	Low
First reply time	Type time duration	Type time duration	Type time duration	Type time duration
Requester wait time	Type time duration	Type time duration	Type time duration	Type time duration
Agent work time	Type time duration	Type time duration	Type time duration	Type time duration
Next reply time	Type time duration	Type time duration	Type time duration	Type time duration
Periodic update	Type time duration	Type time duration	Type time duration	Type time duration
Hours of operation	Calendar hours ▾	Calendar hours ▾	Calendar hours ▾	Calendar hours ▾

Targets allow us to set specific time frames for a range of metrics depending on the priority of the ticket.

For instance, if we want our customers to receive their first reply within three hours while the ticket priority is set to **Low**, we can enter 3 hours in the corresponding field.

While most support systems work with a set priority for each ticket upon its creation, often depending on the content, we set up a system that changes the priority over time. In our case, it may not make sense to set a time target for each priority.

So instead, for ExampleComp, we will set a target only for the **Urgent** priority.

So if a ticket is created, it will progress from the **Low** priority to the **Urgent** priority according to our automations first. If the support is overwhelmed and tickets “become urgent”, our SLAs will start checking if we are breaching our SLAs.

This means that we will need to create two SLA policies. One for EscalationTypeA and one for EscalationTypeB.

Summary

In this chapter, you learned about the different Business Rules Zendesk has to offer. We covered triggers, automations, SLAs, as well as a few different ways to utilize them efficiently in our project. By learning Business Rules, you added a great chunk of knowledge to your repertoire, helping you to understand the inner workings of Zendesk, bringing you closer to our goal of being able to fully customize our environment.

In the next chapter, we will focus on two different aspects: Zendesk's place in a greater ecosystem of tools and the creation of custom Zendesk apps to add more functionality to our setup.

6

Integrating and Extending Zendesk

While Zendesk is a powerful tool for responding to customer enquiries, most likely, it will become a part of a larger eco-system. In order to make the most of our Zendesk environment, we need to integrate Zendesk into our existing systems and extend its capabilities when needed.

Luckily for us, Zendesk allows us to do both of those things.

In this chapter, we will discover Zendesk's extending capabilities by looking at two of the most common integrations – JIRA and Salesforce. Before going into details, we will cover a few more Zendesk basics, including Zendesk apps.

This chapter covers the following topics:

- The Zendesk apps
- Commonly used Zendesk apps from the Marketplace
- Using custom apps to extend Zendesk's capabilities
- Integrating JIRA
- Integrating Salesforce

Zendesk apps

Zendesk apps are a great way to add functionality to our environment. While we can choose from a range of different apps from Zendesk's own marketplace, most companies will choose to create their own custom apps to enable their agents.

In order to gain a better understanding of what Zendesk apps are capable of and how they can be utilized, let's look at a few commonly used apps developed by Zendesk. The best place to find those apps is the Zendesk Marketplace.

Zendesk Marketplace

In order to navigate to the Marketplace, follow these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Marketplace** located under **APPS** within the admin menu:

The screenshot shows the Zendesk Admin interface. On the left, there is a sidebar with various sections: ADMIN HOME, APPS (with Marketplace highlighted and boxed in red), MANAGE (People, User Fields, Organization Fields, Brands, Views, Macros, Reports, Tags, Ticket Fields, Ticket Forms, Dynamic Content, Sandbox), CHANNELS (Email, Twitter, Chat, Facebook, Voice), and a gear icon at the bottom. The main content area has two sections: SYSTEM UPDATES and FEATURE USAGES. The SYSTEM UPDATES section shows three service incidents from October 4th, 2016, with Eduardo Arandilla and Nihar Amin as authors. The FEATURE USAGES section shows metrics for Macros, Triggers, Automations, and Views, along with their usage and update counts for today.

Category	Value	Details
Macros	193	Used today: 124 Updated today: 0
Triggers	107	Used today: 70 Updated today: 1
Automations	20	Used today: 15 Updated today: 0
Views	0	Updated today: 0

Zendesk will open the following web link in your browser: <https://www.zendesk.com/apps/>

Once the page has loaded, we can either use the search bar (if we have an app or keyword in mind) or start by clicking through the categories located at the bottom of the page:

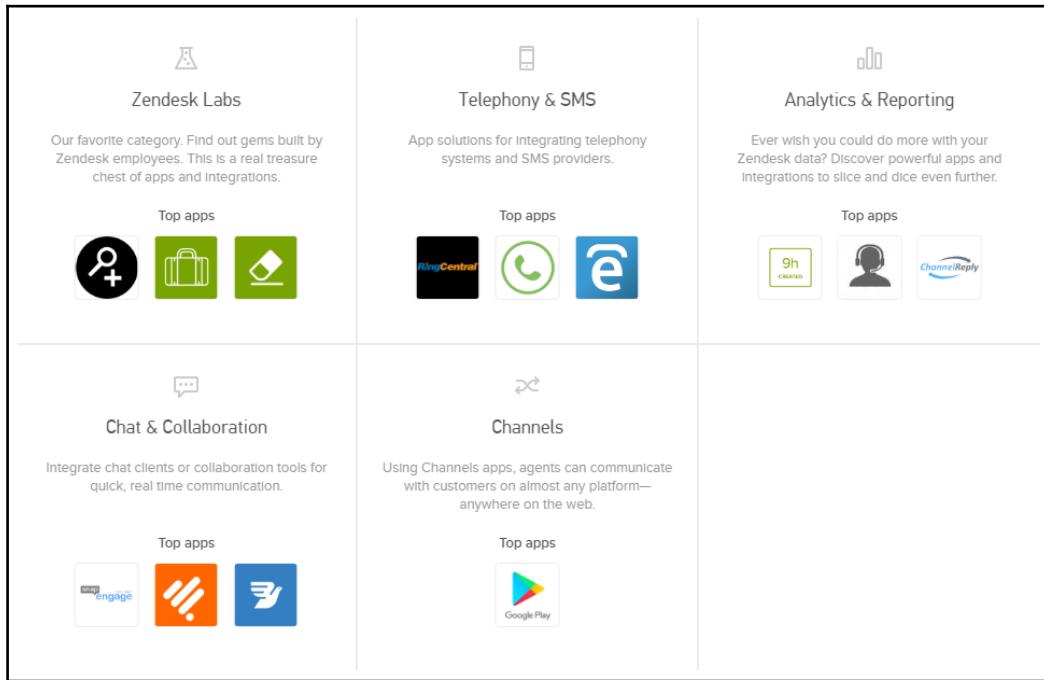
Categories

Zendesk apps come in many different flavors. We have everything from productivity and time tracking, to eCommerce and social media. They're all designed to extend your Zendesk and make life easier for you and your agents.

 Productivity & Time-tracking <p>Our most popular app category features dozen of productivity enhancers, as well as apps to track and organize your time.</p> <p>Top apps</p> 	 E-commerce & CRM <p>Harness Information from your online store or customer database, and use it to do cool stuff.</p> <p>Top apps</p> 	 IT & Project Management <p>An assortment of apps to manage work projects, IT needs, and internal teams.</p> <p>Top apps</p> 
 Knowledge & Content <p>Everything you need to enhance Help Center content and help you find information faster.</p> <p>Top apps</p> 	 Email & Social Media <p>Tools to help with email marketing and managing your social presence.</p> <p>Top apps</p> 	 Surveys & Feedback <p>Products that help you gather feedback about your business and your team's performance.</p> <p>Top apps</p> 

< ... >

Not all categories are visible here. We can scroll through more categories by clicking on the little arrows, as seen in the preceding screenshot.



Zendesk sorts its apps using the following categories:

- **Productivity & Time-Tracking**
- **IT & Project Management**
- **E-Commerce & CRM**
- **Telephony & SMS**
- **Knowledge & Content**
- **Analytics & Reporting**
- **Zendesk Labs**
- **Chat & Collaboration**
- **Surveys & Feedback**
- **Email & Social Media**
- **Channels**

While these categories are very loose and do not always give a clear indication regarding the kind of apps we can expect to find, they help to gain a better understanding about the scope of functionality Zendesk apps have to offer. Let's take a closer look at each one of them.

The **Productivity & Time-Tracking** apps are meant to enhance our team's productivity and give us the option to add time tracking to our environment.

We may, for instance, want to add a time-tracking app in order to get another measurable performance metric.

The **IT & Project Management** apps help us to manage projects by adding team and task management capabilities.

We already know that Zendesk can be used to track tasks, but what if we already use an external tool to do so? If this is the case, we may want to search for the corresponding app here, allowing us to integrate our existing system of choice.

The **E-commerce & CRM** apps can be utilized in order to retrieve information from external systems and their databases.

A lot of companies use CRM tools and have a great amount of customer related data in their databases. It may help our agents to know more about our customer's history when dealing with their enquiry. This category might house the necessary app to make it happen.

The **Telephony & SMS** apps offer integration options for a great range of SMS and telephony providers/systems.

The **Knowledge & Content** apps focus on harnessing information from our Help Center and other systems.

For example, it might be a great help for agents if we could display links to different Help Center articles depending on the ticket's content.

The **Analytics & Reporting** apps make use of our data in order to present us with information that might come in handy for decision making.

Zendesk Labs

The **Zendesk Labs** apps cover a great amount of apps created by Zendesk employees. I do love this category due to the quality of apps available.

The **Chat & Collaboration** apps allow us to integrate chat clients and other tools used for employee collaboration as well as external and internal real-time communication.

The **Surveys & Feedback** apps can be utilized to gather feedback from our customers – a great way to gain more insights into the team's performance and the overall satisfaction of our customers.

The **Email & Social Media** apps allow us to integrate some social media and e-mail the marketing options. If you are looking for a newsletter solution, this might be a good place to start looking.

Channels

The **Channels** apps are great to open channels, allowing our customers to choose almost any platform to communicate with us.

Zendesk app locations

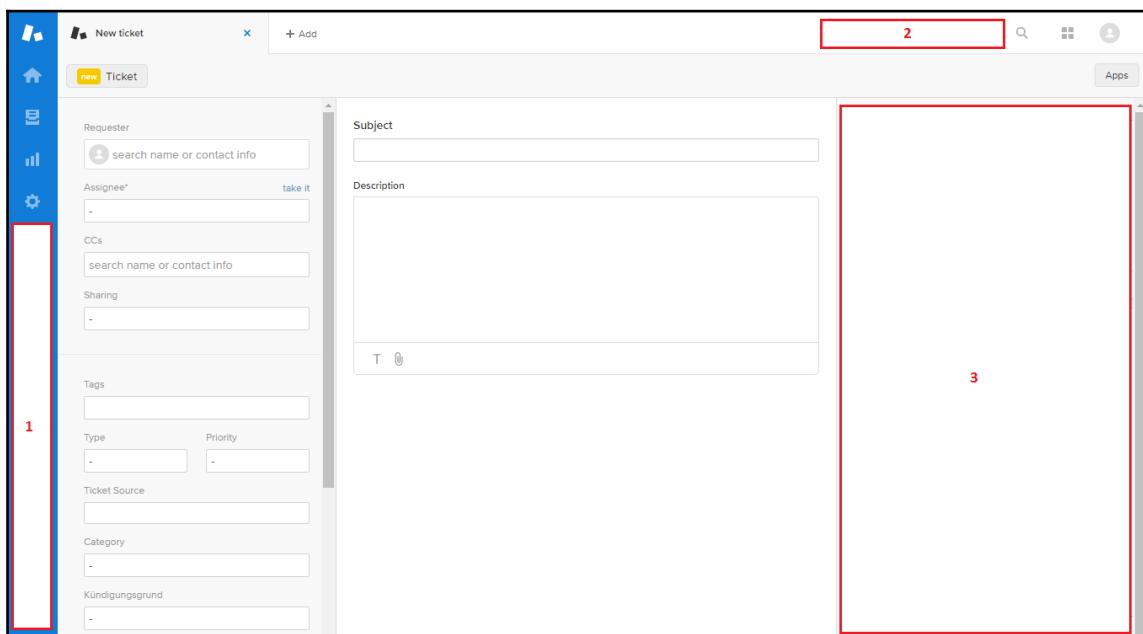
While Zendesk apps can be categorized using the scope of their application, there is another attribute allowing us to divide them into different groups – their location.

But why would we worry about the location of the app? How is this distinction significant enough to even mention it?

Great question! The location of an app stands in direct correlation with the scope of its application. Understanding that can be very helpful when it comes to creating your own apps, or simply for making sense of pre-existing apps in the marketplace.

Before we go into too many details, let's take a look at the possible app locations first. Zendesk apps can be displayed at the following locations:

- Navigation bar on the left-hand side
- Right-hand side of the bar located at the top
- Panel on the right-hand side of the ticket
- Panel on the right-hand side of the new ticket
- Panel on the right-hand side of the user page
- Panel on the right-hand side of the organization page
- No UI always running in the background to receive and react to special events



Apps that are placed in the navigation bar on the left (1) show as icons just like the pre-existing icons within that space. Once the icon is clicked, the now open app is displayed, covering the same space as an open ticket would.

Apps, placed on the right-hand side of the top bar (2), will also show as icons. Once clicked, the app will display as an overlay. The size of this overlay can vary depending on the app.

Apps placed in the panel on the right (3) are usually displayed in full (as in open) as soon as the location is visible to the agent. There are four different locations for this panel:

- When the agent is looking at a ticket (as seen in the screenshot)
- When the agent is creating a new ticket
- When the agent opened a user's profile
- When the agent opened the profile of an organization

Why do we have all these different app locations and what sets them apart?

Apps can be utilized in many different ways. If you wish, for example, to display customer information that would help agents to solve tickets, then it would make sense to place the app in the panel on the right side of an open ticket (3). If this information is important but only in rare cases, you may not want to clutter the ticket panel and move it to the panel located on the user's profile.

If an app has nothing to do with an individual ticket, user, or organization, you may want to place the app either on the navigation bar on the left or the top bar.

In addition to apps with visible UI elements, apps can also only run in the background. This makes sense if the app is waiting for specific events to trigger automated processes.

Zendesk App examples

Let's have a look at some example apps and how to install them. You can find each of the following apps by accessing the Marketplace through Zendesk or by visiting the following website: <https://www.zendesk.com/apps/>

To speed things up, simply copy the name of each app in the search bar.

Show Related Tickets app

The *Show Related Tickets* app was created by Zendesk employees and can be found within the Zendesk Labs category as well:



Show Related Tickets

Identify multiple incidents by surfacing related tickets

★ ★ ★ ★ (499)

INSTALL

Free



APP DETAILS

Author: Zendesk
Support: [Email](#)
Version: 1.0.1

This is an app. It will install right into your Zendesk.

[Twitter](#) [Facebook](#) [Google+](#)

DESCRIPTION

The Zendesk Show Related Tickets app analyzes your ticket subject, searches against all other tickets, and returns any tickets with matching terms.

HOW TO INSTALL

Note: The app will search against only tickets in the "solved" or "closed" states.

The app enables your agents to quickly recognize and identify multiple incidents of the same issue. Then, agents can link all incident tickets to a problem ticket. A resolution to the problem ticket will be applied to all incident tickets, enabling your agents to tackle and resolve issues more efficiently and in a more organized manner.

For any questions please email support@zendesk.com.

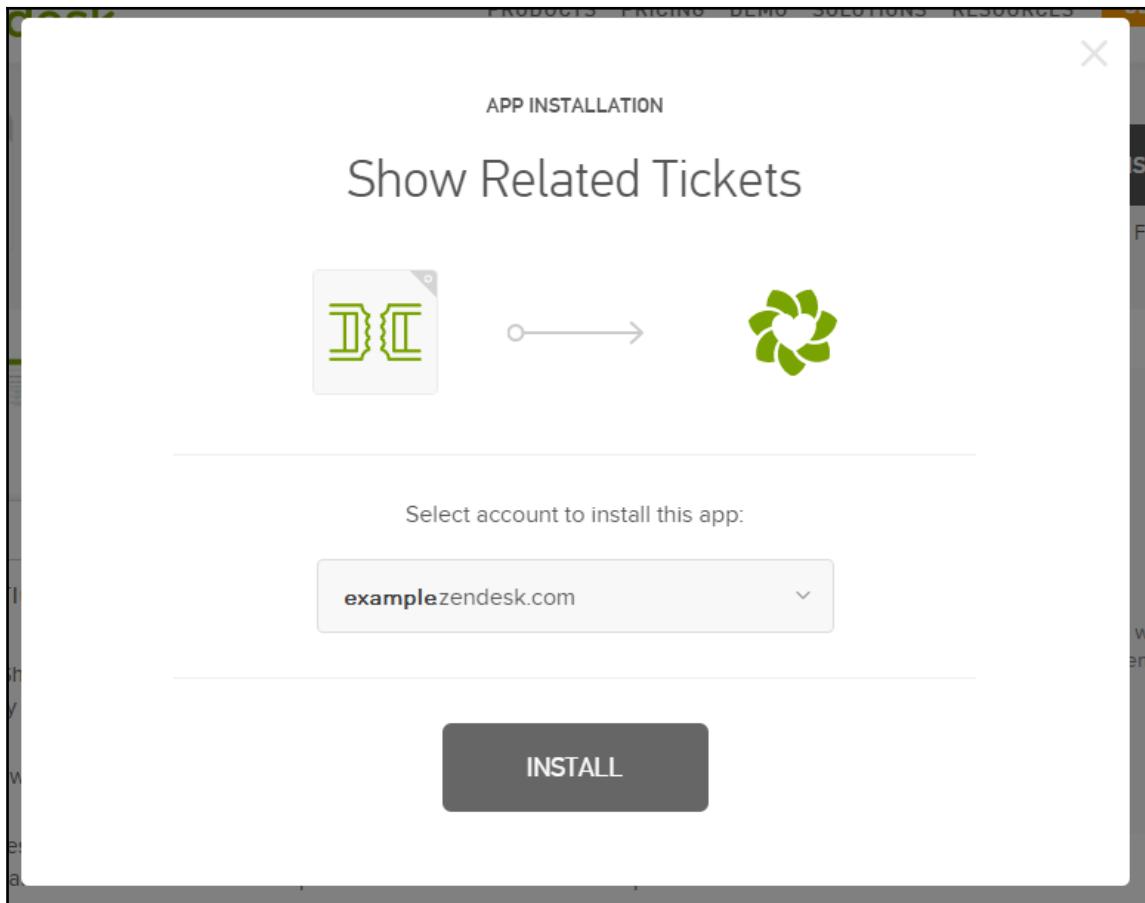
By enabling this app, You agree to the [Built by Zendesk Terms of Use](#).

The Zendesk Show Related Tickets app analyses your ticket subject, searches against all other tickets, and returns any tickets with matching terms.

- Zendesk Marketplace

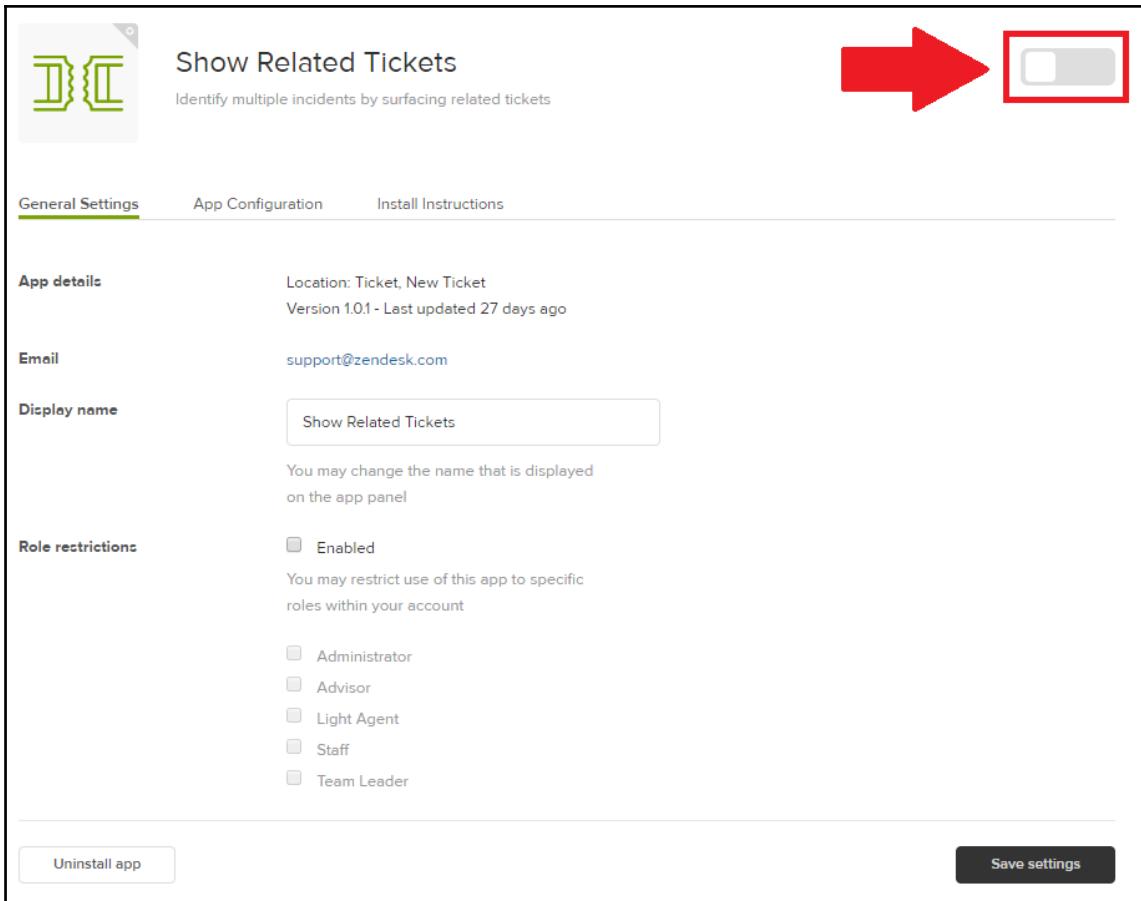
The app is placed within the ticket view and allows agents to find similar issues that have already been set to *solved* or *closed*. Reviewing similar cases can help agents to find previously applied solutions to resolve issues more efficiently.

In order to install the app, we can simply click on the **INSTALL** button and the following pop up will show up:



Make sure the right Zendesk account is selected in the drop down and confirm by clicking on **INSTALL** again.

We will be reverted to our Zendesk environment, showing the app's configuration/ settings page:



In order to enable the app, simply click on the currently grayed out, button in the top-right.
You may switch between the following tabs:

- **General Settings**
- **App Configuration**
- **Install Instructions**

The **General Settings** page will always show the same options and pieces of information, as given here:

- **App details**
- **Email**
- **Display name**
- **Role restrictions**

In this case, the **App details** inform us about the app's location, its current version, and when it has been updated last.

The displayed **Email** can be used in order to contact the developer of the app.

By setting a different **Display name**, we can change the name displayed on the app panel.

Enabling **Role restrictions** allows us to restrict the use of this app to specific roles only.

You can then go on and choose to either save the settings or uninstall the app by clicking on one of the buttons displayed at the bottom, as shown here:



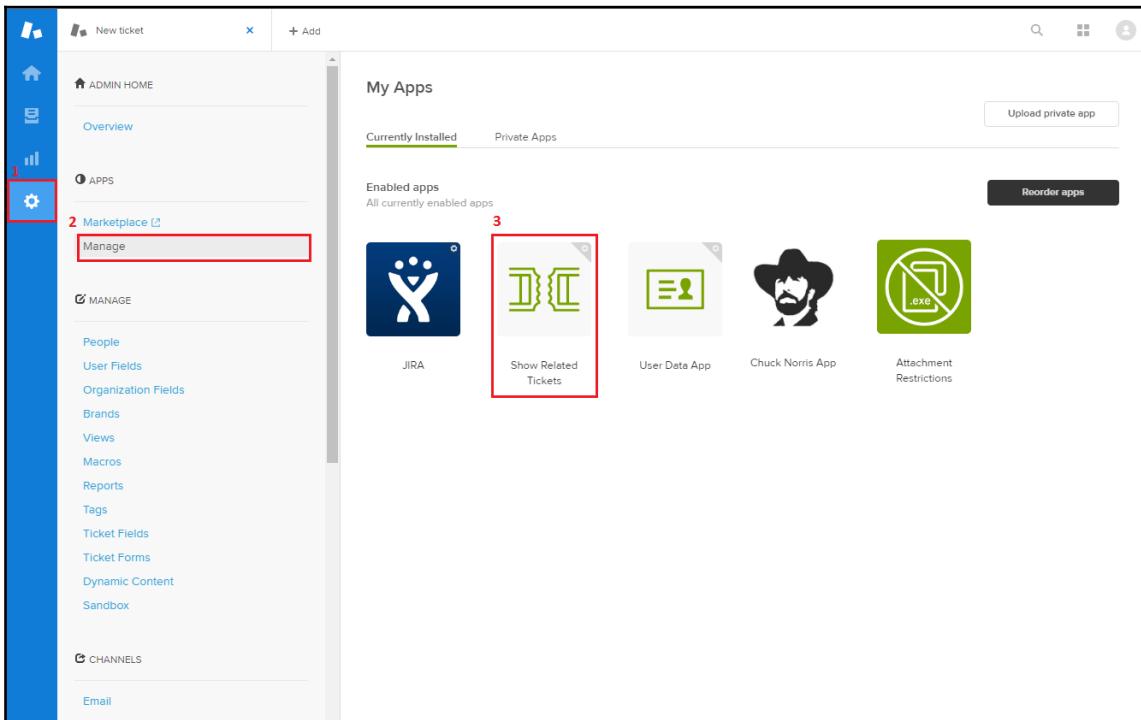
In order to find more app-related settings, switch to the **App Configuration** tab.

While the installation of this app was pretty straightforward, some apps may need additional steps to finalize the setup. You can find a detailed list by switching to the **Install Instructions** tab.

That is it. In order to check the app, simply switch to a ticket.

If you wish to change the settings later on, you can always do so by following these steps:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Manage** located under **APPS** within the admin menu.
3. Click on the app icon of your choice:



Iframe app

The *Iframe* app was also created by Zendesk employees and can therefore be found within the **Zendesk Labs** category as well:

The screenshot shows the Zendesk Marketplace page for the "Iframe" app. At the top, there's a large icon of a computer monitor with a green arrow pointing from one window to another. To the right of the icon, the word "Iframe" is written in a large, bold, black font. Below the title, a subtitle reads "Embed any website in your Zendesk (formerly known as Sidebar Icon app)". A dark grey "INSTALL" button is on the right, with the word "Free" underneath it. Below the title, there's a row of five small stars followed by "(0)". On the far right, there's a "DESCRIPTION" tab and a "HOW TO INSTALL" tab. The "DESCRIPTION" tab is currently active, showing a paragraph about what the app does. To the right of the description, there's a section titled "APP DETAILS" with information about the author (Zendesk), support (Email), and version (1.0.2). Below the "APP DETAILS", there's a note stating "This is an app. It will install right into your Zendesk." At the bottom right, there are social sharing icons for Twitter, Facebook, and Google+.

An *Iframe* is an HTML element that lets you nest one web page in another web page. The Zendesk Iframe app lets your agents quickly navigate to an external website and then return to their tickets without interrupting their workflow. Your agents can look up any external site and reference information such as order details or data on another system.

The Iframe app presents the external source as an iframe in your Zendesk. Your agents can access this page by clicking the Iframe app icon.

The Iframe app supports websites that include the Zendesk App Framework (ZAF) SDK. Check out [Iframes in Apps](#) to learn more.

See this [Iframe app article](#) for more information on how to setup the app.

For questions, please email support@zendesk.com.

By enabling this app, You agree to the [Built by Zendesk Terms of Use](#).

The Iframe app presents the external source as an Iframe in your Zendesk. Your agents can access this page by clicking the Iframe app icon.

– Zendesk Marketplace

Sounds good, but what exactly is an Iframe?

An Iframe is an HTML element that lets you nest one web page in another web page.

– Zendesk Marketplace

Before we can proceed, we will need to take care of the following settings:

And why would we need such a feature?

The Zendesk Iframe app lets your agents quickly navigate to an external website and then return to their tickets without interrupting their workflow. Your agents can look up any external site and reference information such as order details or data on another system.

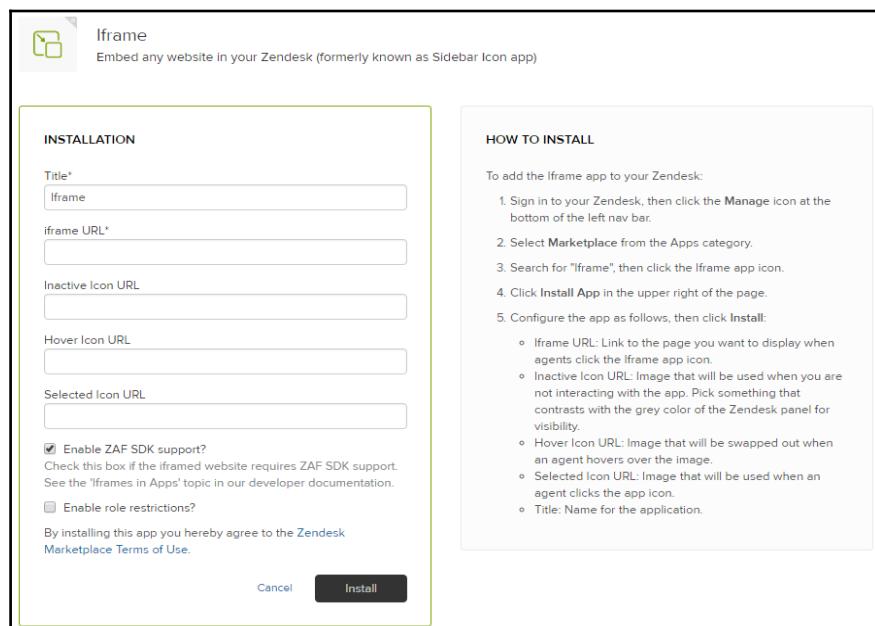
– Zendesk Marketplace

As you can see, I was able to answer the most pressing questions by quoting the app description itself. This is why I like browsing the app store from time to time. You never know what you will find. Often enough, an app gives me a great idea for my own projects and custom apps.

In order to install the app, let's review the instructions first by clicking on **HOW TO INSTALL**.

Again, the process seems pretty straightforward, so let's click on **INSTALL**. Make sure to select the right Zendesk account within the pop up's drop down and click on **INSTALL** again.

As soon as we are referred back to Zendesk, we are presented with some options:



Before we can proceed, we will need to take care of the following settings:

- **Title**
- **iframe URL**
- **Inactive Icon URL**
- **Hover Icon URL**
- **Selected Icon URL**
- **Enable ZAF SDK support?**
- **Enable role restrictions?**

1. First of all, we will need to come up with a **Title** for our application. Since the title should correlate with the purpose of the app, let's assume we are using the app to provide quick access to ExampleComp's internal wiki page. In that case, the title could simply be `Wiki`.
2. Next, we will have to set **iframe URL**, which should reflect the URL of our `wiki`, which would look something like this:

`https://wiki.examplecomp.com/`

3. Then we will set **Inactive Icon URL**, **Hover Icon URL**, and **Selected Icon URL**. Since this app is located as an icon within our navigation bar on the left, we will need to supply these icon files.



The size of such icons (for the navigation bar) should be 35×35 pixels. Zendesk also suggests some padding on the sides, to primarily use `#FFF` for color, to use 10% opacity for the inactive state (inactive icon), and 60% for the active state (selection icon). You may want to forward these information to the department in charge.

Once the icons have been prepared, you may want to upload them on a company server and use the corresponding URLs.

4. Next, we are asked to decide whether **Zendesk App Framework SDK (ZAF SDK)** support should be enabled. This will not be needed in our case, but might be a great option for some cases.

You can embed websites in your apps with iframes. The Apps framework provides a set of APIs that allow you to post and receive messages from both your website and your app.

- Zendesk

5. Last but not least, we can select **Enable role restrictions** and confirm by clicking on **Install**.

That is it. The icon should show up on your navigation bar on the left. Try it out by clicking on it.

Out of Office app

Again we are looking at an app created by Zendesk employees:

The screenshot shows the Zendesk Marketplace page for the "Out of Office" app. At the top, there's a green icon of a briefcase, the app name "Out of Office", a brief description "Keep track of, prepare for, and handle out of office agents and their tickets", a rating of 4.5 stars with 617 reviews, and a large "INSTALL" button. To the right of the button is the word "Free". Below this, there are three screenshots of the app's interface showing various ticket management features. At the bottom of the main section are two tabs: "DESCRIPTION" (which is currently selected) and "HOW TO INSTALL". The "DESCRIPTION" tab contains sections for "Purpose" (described as "Keep track of, prepare for, and handle out of office agents and their tickets") and "Details" (described as "This app will allow Admins to manage their Agents' vacation statuses, as well as allowing individual Agents to manage their own vacation statuses, and reassigning assigned tickets to their parent group if updated during the vacation"). It also notes that there's an option to unassign open tickets when setting an Agent on vacation. To the right of the main content is a "APP DETAILS" sidebar with information: Author: Zendesk Labs, Support: Not supported, Version: v2.6.3, and a note that it's an app that installs directly into Zendesk. Below this are social sharing icons for Twitter, Facebook, and Google+.

Keep track of, prepare for, and handle out of office agents and their tickets.

- Zendesk Marketplace

This app is a little more complex and greatly demonstrates what Zendesk apps are capable of. It also serves as a great learning experience and I suggest that you read the instructions by clicking on **HOW TO INSTALL** first.

Try installing the app and test its features by playing around with it for a while. You may find it helpful for your Zendesk setup in the future.

Creating custom Zendesk apps

Using apps from the Marketplace can make a real difference to our Zendesk experience. Though, at some point, you may need to create your own apps in order to reach your goals.

It is important to note, in order to create your own apps, you will need to have some pre-existing knowledge of the following:

- HTML
- JavaScript
- CSS

However, even if such topics are new to you, a better understanding regarding the development of Zendesk apps can be very helpful when creating such apps with the help of seasoned developers. So let's go through the process step by step.

Planning a custom app

Let's stick to our road map and remember the following:

If we want to know more about our customers while answering their requests, it would be helpful to create our own little Zendesk app to help us out. Most companies have a unique identifier for each customer. In our case, we could use the customer's e-mail address. We could either display a generated link, that would open the companies backend, displaying the necessary information, or we could go one step further and query that information from our servers and then display it within our ticket view itself. Chapter 1, Configuring Your Own Zendesk.

To keep this as simple as possible, we will go with option one and create an app that would generate such a link. Before we start, let's go through the app's inner workings:

1. Get the requester's e-mail address.
2. Generate the link to the requester's backend page.
3. Display the link within our app.

Installing the necessary tools

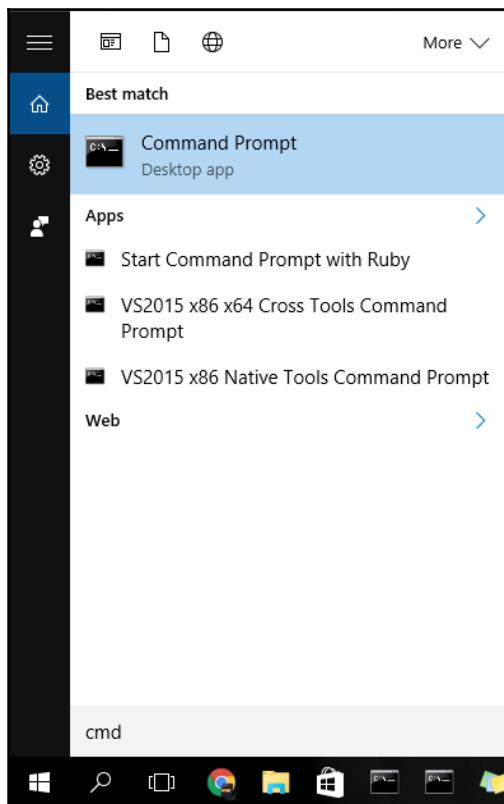
Before we can start creating our own app, we will need to install the necessary tools called **Zendesk App Tools (ZAT)**, which let you do a couple of things. Most importantly, they will allow the creation of the files and folders needed for a new app as well as the option to test your app locally during development.

The ZAT are a Ruby gem. While Ruby is a programming language, a Ruby gem is a standard format for Ruby programs and libraries capable of extending Ruby's functionality.

Installing Ruby

Let's start by checking whether Ruby has been installed on your system previously:

1. First, we will need to open the **Command Prompt** by entering `cmd` in the Windows search box, followed by clicking on the top search result:



2. Next, enter the following command and press *Enter*:

ruby -v



If you are using a Mac, you can utilize the same command within your Mac's **Command Prompt**.

The screenshot shows a standard Windows Command Prompt window. The title bar says "CMD". The main area shows the command "C:\>ruby -v" entered at the prompt. The window has a scroll bar on the right side.

This command is meant to display the version of your current Ruby install. If you receive an error message instead, we can assume that you have not installed it previously and you will need to proceed by installing Ruby now.

Installing Ruby in Windows

You can download the latest version for Windows by visiting the following web link:

<http://rubyinstaller.org/downloads/>

Installing Ruby in Mac

In order to install Ruby on a Mac, I would suggest that you use the all-in-one Mac installer from RailsInstaller. While it includes a variety of other tools as well, it is one of the safer methods to avoid any errors. You can download RailsInstaller from the following link:
<http://railsinstaller.org/en>

Installing the ZAT gem

In order to install the necessary tools, enter the following command in your Command Prompt (Windows or Mac):

```
gem install zendesk_apps_tools
```

The installation might take a few moments.

Creating app files

Next, we can start creating our app. Luckily, we can create an empty project utilizing our Zendesk App Tools, by following these steps:

1. Create a folder for your Zendesk apps (Example: C:\Zendesk Apps\).
2. Navigate to this folder within the **Command Prompt**:

```
cd C:\Zendesk Apps
```

3. Enter the following command:

```
zat new
```

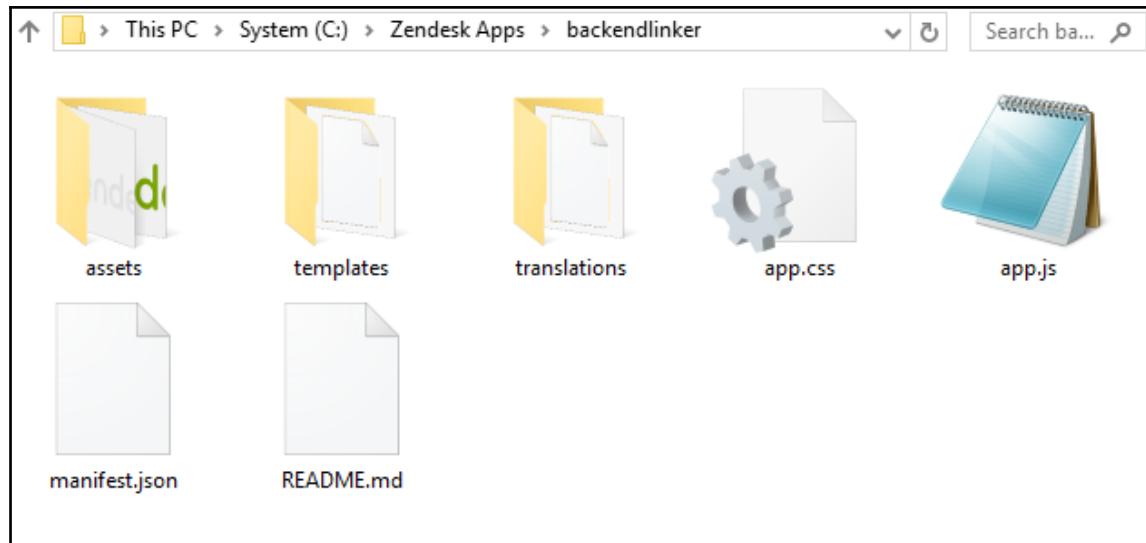
4. Follow the prompts:

If done correctly, it will look something like this:

```
C:\>cd C:\Zendesk Apps
C:\Zendesk Apps>zat new
Enter this app author's name:
Cedric Jacob
Enter this app author's email:
contact@cedricfjacob.com
Enter this app author's url:
www.cedricfjacob.com
Invalid url, try again:
Enter this app author's url:
https://www.cedricfjacob.com
Enter a name for this new app:
backendlinker
Enter a directory name to save the new app (will create the dir if it does not exist, default to current dir):
backendlinker
  create backendlinker
  create backendlinker/README.md
  create backendlinker/app.css
  create backendlinker/app.js
  create backendlinker/assets/banner.png
  create backendlinker/assets/logo-promotion.png
  create backendlinker/assets/logo-small.png
  create backendlinker/assets/logo.png
  create backendlinker/manifest.json
  create backendlinker/templates/layout.hbs
  create backendlinker/translations/en.json

C:\Zendesk Apps>
```

As you can see, I made a little mistake at first. Not to worry though, we will receive an explanatory error message, but keep going. Next, open your explorer and navigate to the newly created files:



Let's go through the created files really quick:

- The assets folder:
 - banner.png
 - logo.png
 - logo-promotion.png
 - logo-small.png
- The templates folder:
 - layout.hdbs
- The translations folder:
 - en.json
- app.css
- app.js
- manifest.json
- README.md

The assets folder will hold all the asset files for your app. It already holds some of the standard image files needed for your app, and you can simply replace them as you see fit:



The templates folder holds template files, which define the look of your app. By default, the template is named `layout.hdbs`. You can use any text editor to open the file and review its content, as shown here:

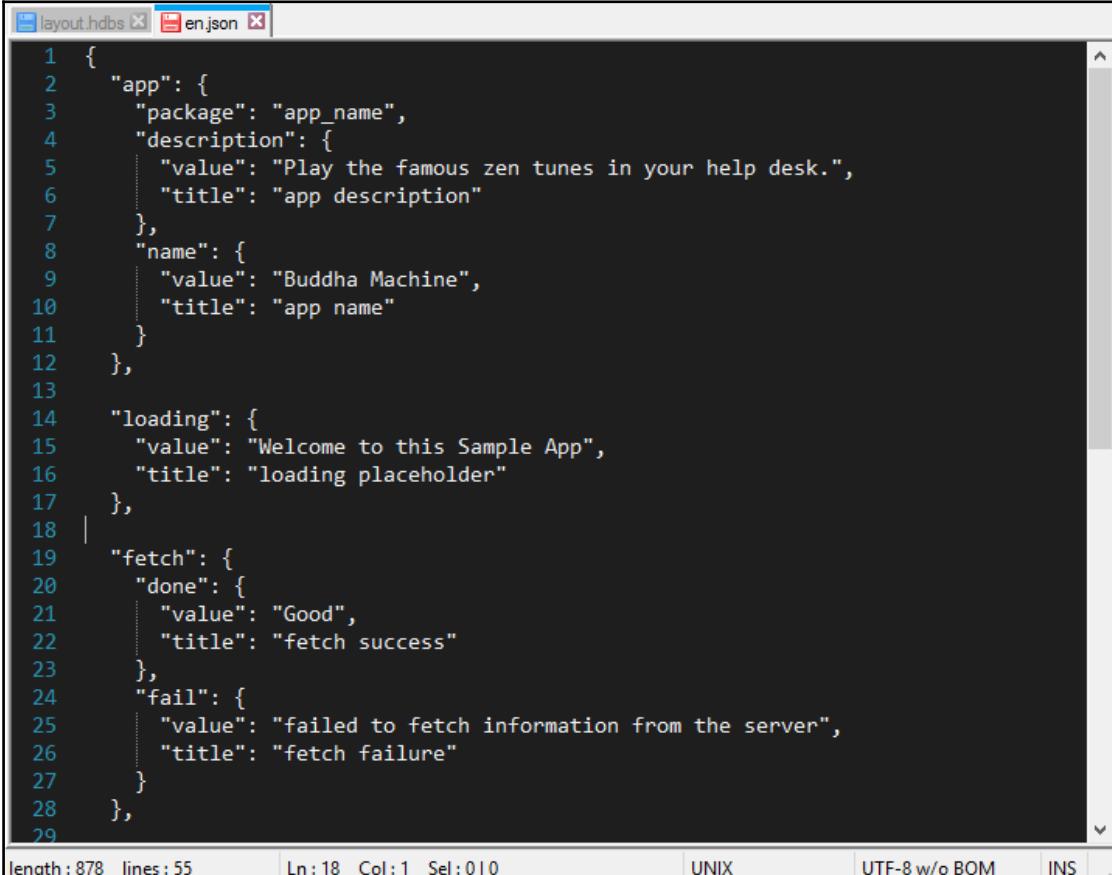
```
layout.hdbs
1 <header>
2   <span class="logo"/>
3   <h3>{{setting "name"}}</h3>
4 </header>
5 <section data-main/>
6 <footer>
7   <a href="mailto:{{author.email}}">
8     {{author.name}}
9   </a>
10 </footer>
11
```

Ln : 11 Col : 1 Sel : 0 | 0 UNIX UTF-8 w/o BOM INS ...

A screenshot of a code editor window titled 'layout.hdbs'. The code is written in HTML-like syntax. Lines 1 through 11 are visible. Line 1 starts with '<header>'. Line 2 contains ''. Line 3 contains '<h3>{{setting "name"}}</h3>'. Line 4 ends the header section with '</header>'. Line 5 starts a new section with '<section data-main/>'. Line 6 starts a footer section with '<footer>'. Lines 7 through 9 are inside the footer section, starting with '' and ending with ''. Line 10 ends the footer section with '</footer>'. Line 11 is a blank line. At the bottom of the editor, there are status bars showing 'Ln : 11', 'Col : 1', 'Sel : 0 | 0', 'UNIX', 'UTF-8 w/o BOM', 'INS', and an ellipsis button.

This is where your **HTML (Hypertext Markup Language)** knowledge comes in handy. You may add and remove elements as you please, but for now, let's move on.

The `translations` folder holds the translation files, which work similar to dynamic content. Let's take a look at the `en.json` file:



A screenshot of a code editor window titled "layout.hdbs" showing the "en.json" file. The code is a JSON object with several properties and nested objects. The properties include "app", "loading", and "fetch". The "app" property contains "package" and "description" objects. The "loading" property contains "value" and "title" objects. The "fetch" property contains "done" and "fail" objects. Each object has a "value" and "title" key. The code editor interface includes tabs for "layout.hdbs" and "en.json", status bar with file length (878), lines (55), and cursor position (Ln: 18 Col: 1 Sel: 0 | 0), and toolbars for UNIX, UTF-8 w/o BOM, INS, and other file operations.

```
1  {
2    "app": {
3      "package": "app_name",
4      "description": {
5        "value": "Play the famous zen tunes in your help desk.",
6        "title": "app description"
7      },
8      "name": {
9        "value": "Buddha Machine",
10     "title": "app name"
11   }
12 },
13
14 "loading": {
15   "value": "Welcome to this Sample App",
16   "title": "loading placeholder"
17 },
18 |
19 "fetch": {
20   "done": {
21     "value": "Good",
22     "title": "fetch success"
23   },
24   "fail": {
25     "value": "failed to fetch information from the server",
26     "title": "fetch failure"
27   }
28 },
29 }
```

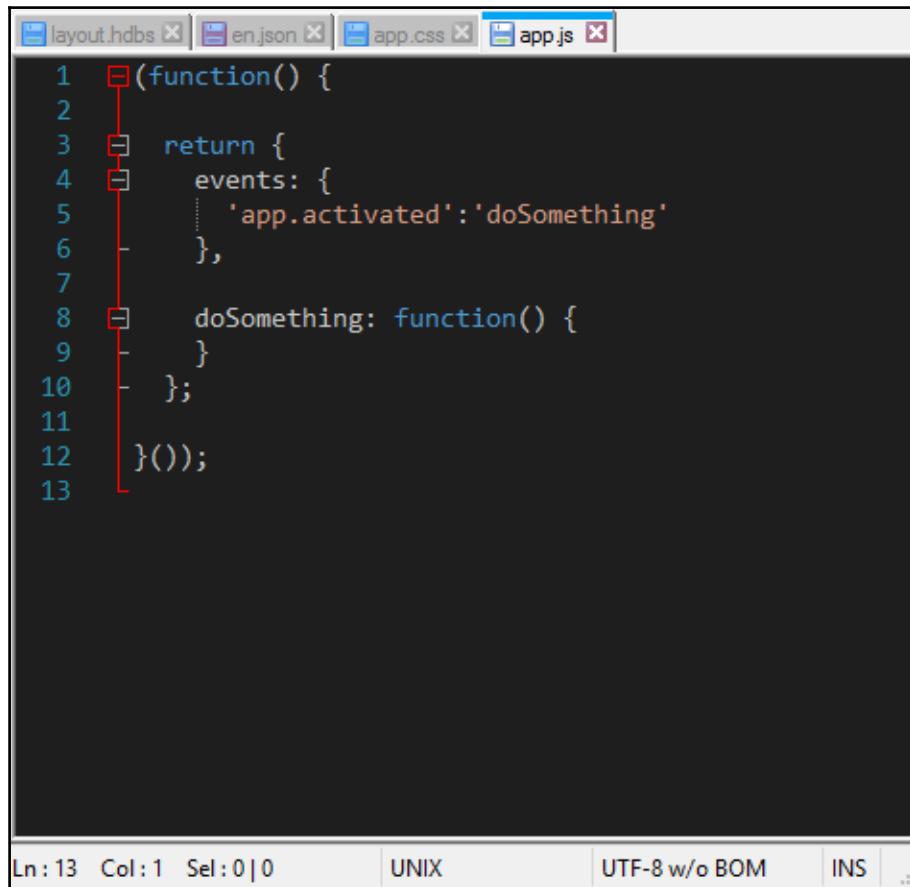
length: 878 lines: 55 Ln: 18 Col: 1 Sel: 0 | 0 UNIX UTF-8 w/o BOM INS ...

The `en.json` file lets us define placeholders and their English translations. We can copy and rename this file for other languages as well. For now, we will not worry about creating an app supporting multiple languages.

Next up, we have `app.css`. Again, we can open this file with any text editor. **CSS (Cascading Style Sheets)** is a style sheet language.

While this file is empty, we can use it to style our HTML elements from our layout file.

The `app.js` file holds the JavaScript for our app. JavaScript is an object-oriented programming language. Here is what the pregenerated content looks like:



A screenshot of a code editor window titled "app.js". The window shows the following JavaScript code:

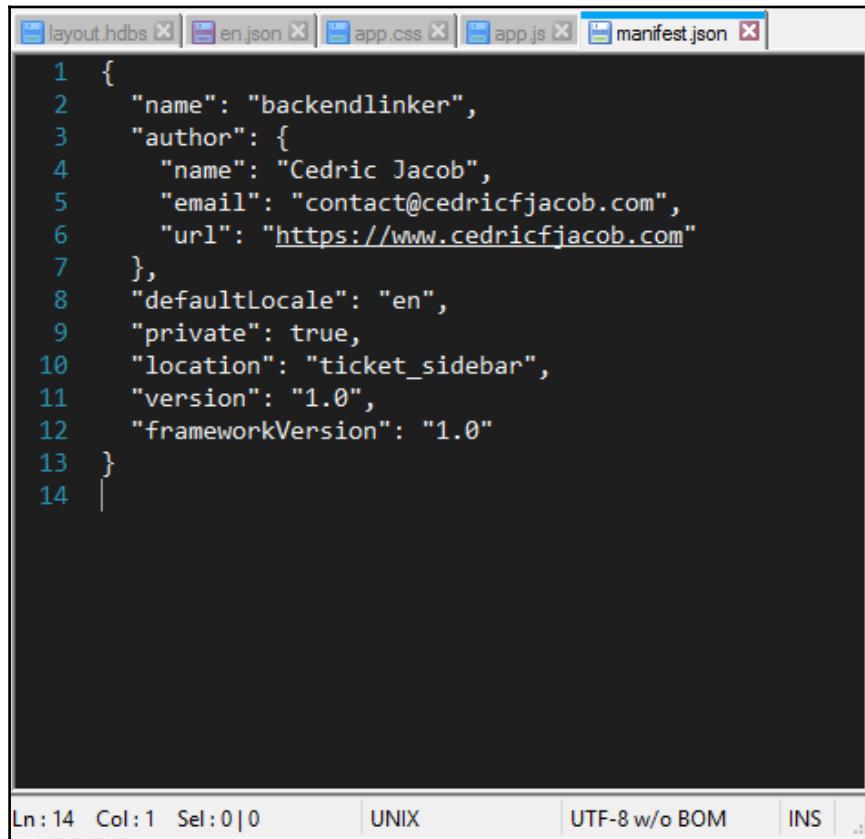
```
1 (function() {
2
3     return {
4         events: {
5             'app.activated': 'doSomething'
6         },
7
8         doSomething: function() {
9             ...
10        }
11    }();
12
13 })();
```

The code defines an anonymous function that returns an object. This object contains an `events` property which maps the event `'app.activated'` to the string `'doSomething'`. It also contains a `doSomething` method. The entire function is wrapped in parentheses and immediately invoked at the end.

At the bottom of the editor, status bars show "Ln: 13 Col: 1 Sel: 0 | 0", "UNIX", "UTF-8 w/o BOM", "INS", and a small icon.

We can see an event called `app.activated`, which will be triggered as soon as the app starts. We can also see a function named `doSomething`, which will be called as soon as the `app.activated` event is triggered. We will use this function later on.

Upon opening the `manifest.json` file, we recognize a few of our previously supplied information that we entered when creating the app files using ZAT:



The screenshot shows a code editor window with several tabs at the top: `layout.hbds`, `en.json`, `app.css`, `app.js`, and `manifest.json`. The `manifest.json` tab is active. The main pane displays the following JSON code:

```
1  {
2    "name": "backendlinker",
3    "author": {
4      "name": "Cedric Jacob",
5      "email": "contact@cedricfjacob.com",
6      "url": "https://www.cedricfjacob.com"
7    },
8    "defaultLocale": "en",
9    "private": true,
10   "location": "ticket_sidebar",
11   "version": "1.0",
12   "frameworkVersion": "1.0"
13 }
14 |
```

At the bottom of the editor, there are status indicators: `Ln: 14 Col: 1 Sel: 0 | 0`, `UNIX`, `UTF-8 w/o BOM`, and `INS`.

While the content is almost self-explanatory, let's take a quick look at the `location` option. We remember the following:

The location of an app stands in direct correlation with the scope of its application. Understanding that can be very helpful when it comes to creating your own apps or simply for making sense of pre-existing apps in the marketplace.

- Zendesk App Locations

While we covered possible locations before, this is how we can define the `location` option in our manifest file:

```
"location": "ticket_sidebar"
```

The possible values are as follows:

- `top_bar`
- `nav_bar`
- `ticket_sidebar`
- `new_ticket_sidebar`
- `user_sidebar`

For our purpose, `ticket_sidebar` is just fine and we can leave it as that.

The `README.md` file contains an example `readme`-file. If we wanted to distribute and sell our app in Zendesk's marketplace, it would make sense to provide proper instructions and help for customers.

Previewing our app

Before creating our own app, you should learn how to test/preview our app without having to upload and install it every single time.

Luckily, our ZAT can help us with that. Simply follow these steps:

1. Navigate to our app's folder using the command prompt:

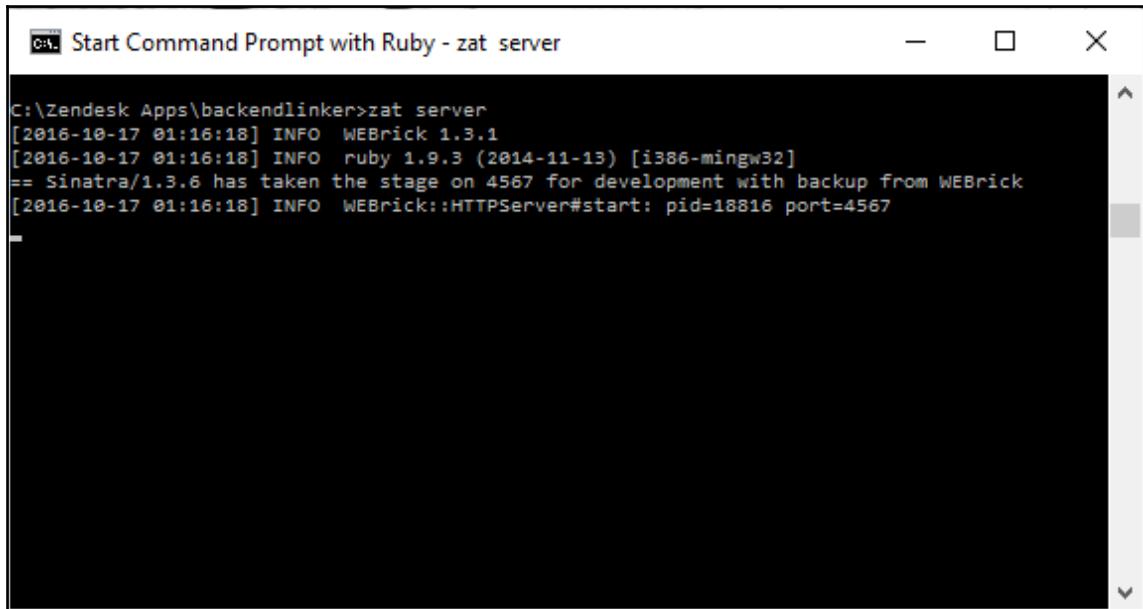
```
cd C:\Zendesk Apps\backendlinker
```

2. Enter the following command (do not close the command prompt after this step):

```
zat server
```

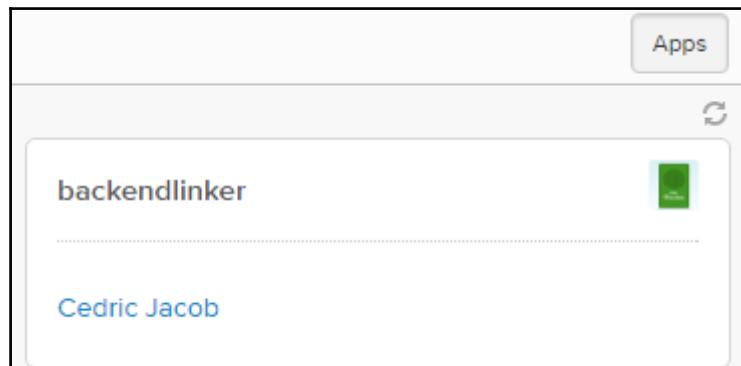
3. Open Zendesk within your browser and navigate to a ticket.
4. Append `?zat=true` to your ticket URL.
5. Depending on the browser, you may see a little shield icon in the address bar.
Click on it and allow your browser to load unsafe scripts.

The command prompt should look something like this:



```
C:\Zendesk Apps\backendlinker>zat server
[2016-10-17 01:16:18] INFO  WEBrick 1.3.1
[2016-10-17 01:16:18] INFO  ruby 1.9.3 (2014-11-13) [i386-mingw32]
== Sinatra/1.3.6 has taken the stage on 4567 for development with backup from WEBrick
[2016-10-17 01:16:18] INFO  WEBrick::HTTPServer#start: pid=18816 port=4567
```

Also, our app in Zendesk should look similar to this:



Well done! It is time to get started on the app itself.

Coding our app

Let's start with creating the layout by editing `layout.hdbs`. This is what our code should look like at this point:

```
<header>
  <span class="logo"/>
  <h3>{{setting "name"}}</h3>
</header>
<section data-main/>
<footer>
  <a href="mailto:{{author.email}}">
    {{author.name}}
  </a>
</footer>
```

We may want to start by cleaning the code a little. In order to do that we will do the following:

1. Remove the logo.
2. Hardcode the displayed name.
3. Remove the footer.

Doing that will reduce our code to the following lines:

```
<header>
  <h3>Backend Linker</h3>
</header>
<section data-main/>
```

Next, we will add the element that will display our link to the backend:

```
<header>
  <h3>Backend Linker</h3>
</header>
<section data-main/>
<a href="" id="backend_href_element" target="_blank">Open Backend</a>
```

When reloading our Zendesk page, the app should look way more sleek:



So far so good. Our app displays the link. Next, we will edit the `app.js` file to add the necessary magic needed to make the link work as planned. This is what the code should look like:

```
(function() {  
  
    return {  
        events: {  
            'app.activated':'doSomething'  
        },  
  
        doSomething: function() {  
        }  
    };  
  
}());
```

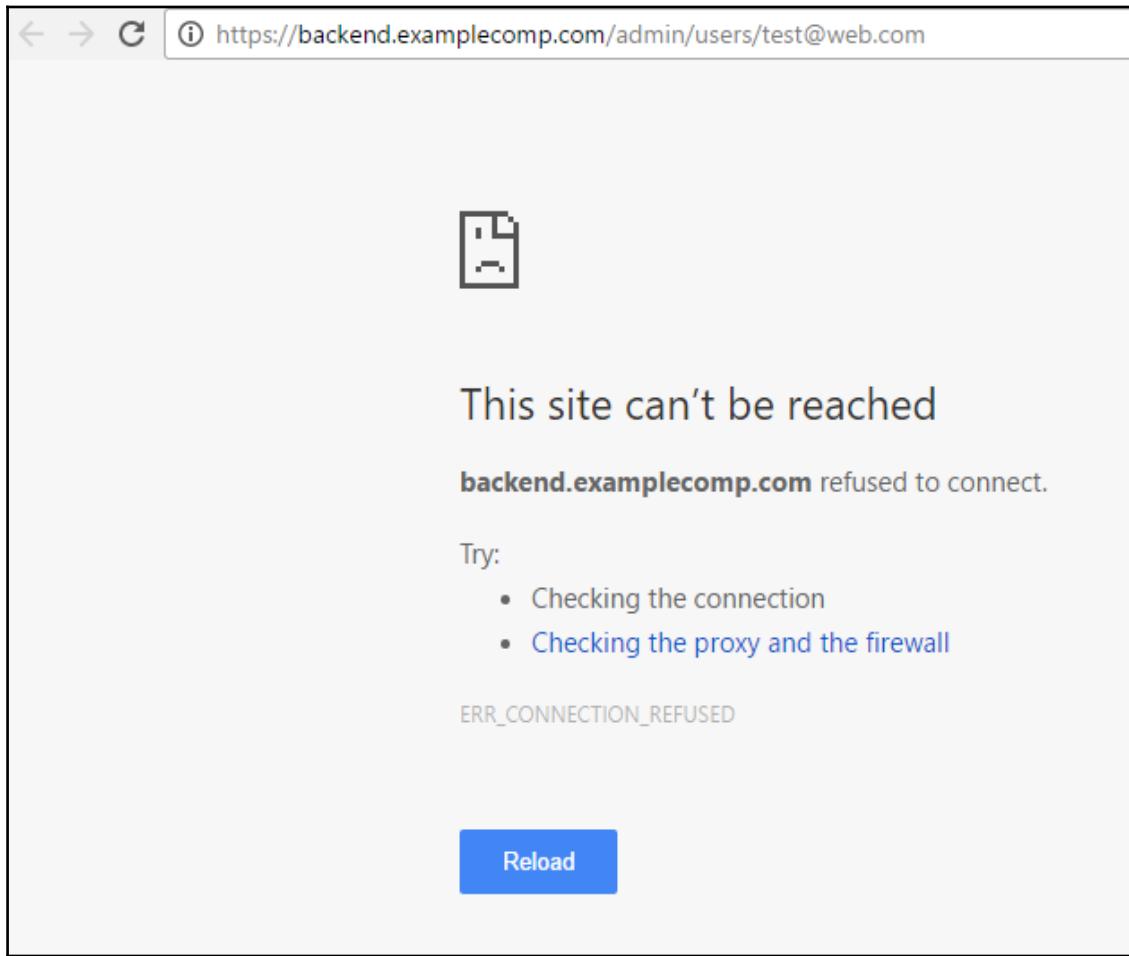
We can focus on the `doSomething` function, which will be called as soon as the app is activated. But what do we need to do again? We need to do the following:

1. Obtain the requester's e-mail address.
2. Generate the right link.
3. Manipulate the link in our layout.

This is what our new function should look like:

```
doSomething: function() {  
    //Use Data API to obtain email address of requester  
    var email = this.ticket().requester().email();  
    //Generate the link  
    var link ='https://backend.examplecomp.com/admin/users/'  
    + email;  
    //Manipulate link in layout  
    this.$("#backend_href_element").attr("href", link);  
}
```

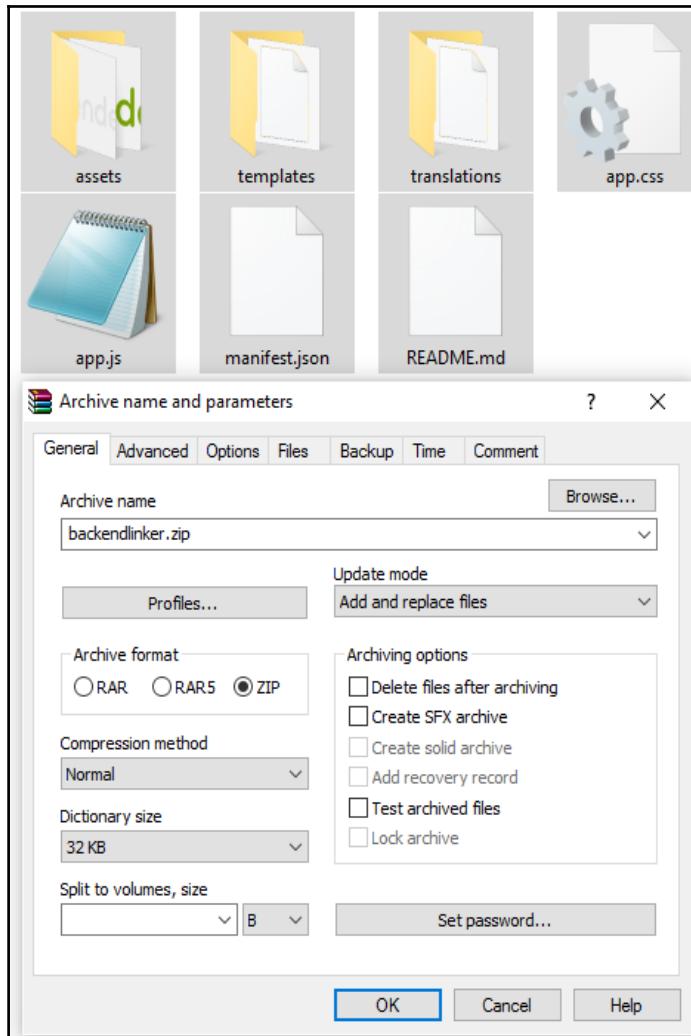
After saving the `app.js` file, we can reload our browser to display the updated app. When clicking on the displayed link, a new tab should open:



Our app worked! Unfortunately, ExampleComp is a made up company and there is no backend that would display any account management options for `test@web.com`.

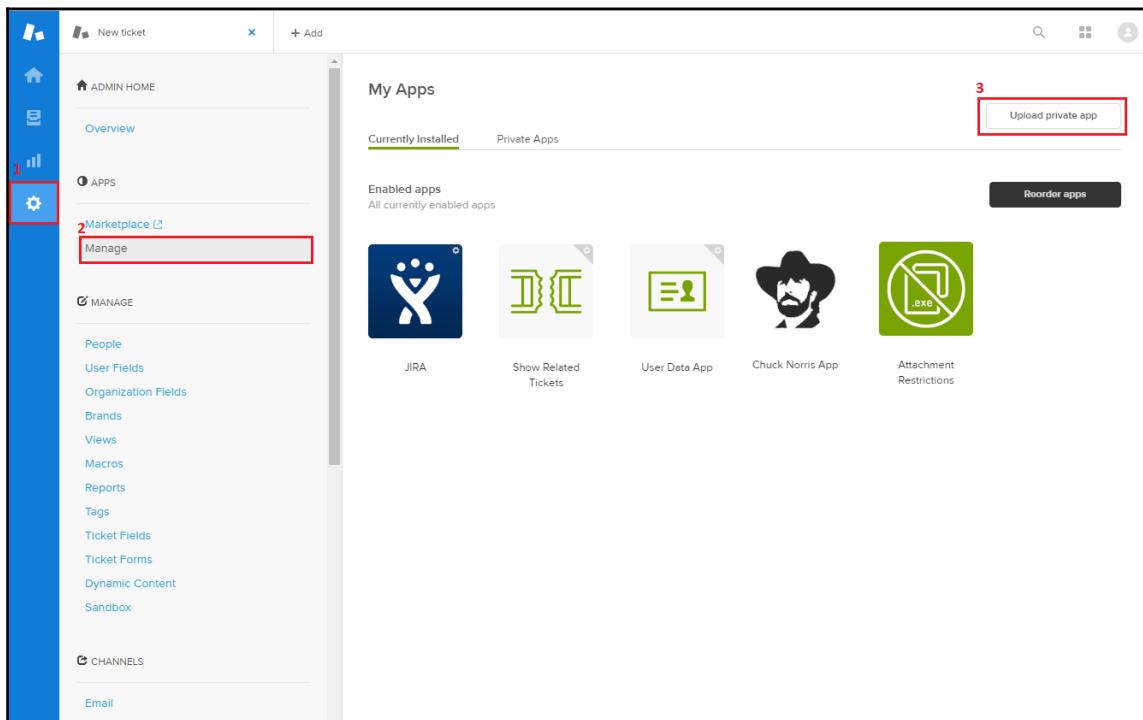
Uploading/installing our app

Before we can upload our new creation, we will need to pack our app. You can use any compression tool that allows creating ZIP files. I personally use WinRAR:



Follow these steps in order to upload the app:

1. Click on the Admin icon (gear symbol), located at the bottom of Zendesk's sidebar.
2. Click on **Manage**, located under **APPS** within the admin menu.
3. Click on **Upload private app** in the top-right corner:



Zendesk will prompt us to enter the app's name and to choose a local file from our hard drive:

Upload App

This is where you upload your private apps. Private apps appear in your account only. If you want to submit a public app or learn more about how to build your own private app, check out our [Developer documentation](#).

App Name

Blank

This is seen when you list your app. Please follow the [naming guidelines](#).

App File

Choose File

Must be a .zip file. Max file size 2mb.

Do not include any secret passwords, keys, or tokens in your app's assets directory. Access to these files is not authenticated. See [Secure Settings](#) for more information.

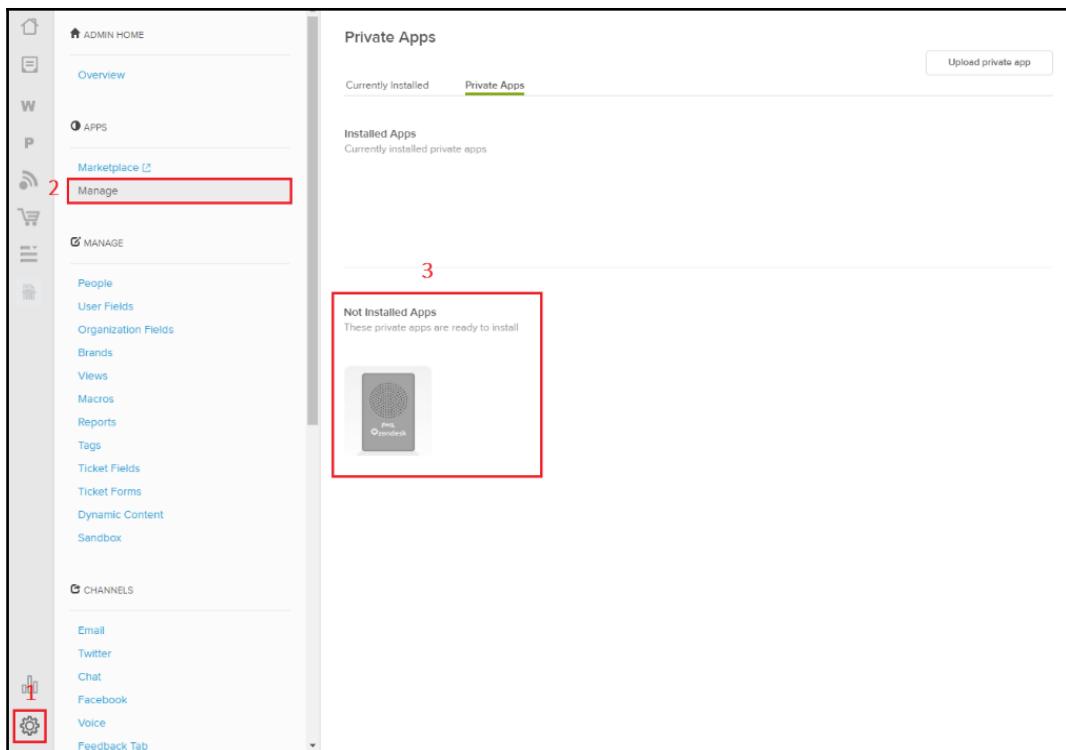
[Cancel](#)

[Upload](#)

After uploading the ZIP file, we will have to install the app next. Follow these steps in order to install the app:

1. Click on the Admin icon (gear symbol), located at the bottom of Zendesk's sidebar.
2. Click on **Manage**, located under **APPS** within the admin menu.
3. Click on the app located under **Not Installed Apps** and select **Install** from the context menu.

4. Follow the installation process.



Well done! You just installed your first custom-made Zendesk app.

Really, we only scratched the surface. Creating complex Zendesk apps can be a lot of fun and a rewarding process.

If you would like to learn more about creating Zendesk apps, I would suggest that you review Zendesk's documentation available at <https://developer.zendesk.com/apps/docs/>.

JIRA integration

Integrating JIRA is not as simple as installing an app from the Marketplace. In order to understand the whole process, you should have some experience with JIRA or (and this is how I prefer to do it) work together with someone in charge of the JIRA setup.

Integration preparations

Before we can start, we need to make sure that a few conditions are met:

- You must have admin rights in both Zendesk and JIRA
- The JIRA account is both in the jira-administrators and the jira-software-users group
- The reporter field has to be present on the **Create Issue** screen
- Third-party cookies are enabled in your browser
- Atlassian add-ons need the permission to access JIRA projects

The first condition is a no-brainer. Without having the admin permissions, we simply cannot make the necessary changes for this to work. It might be noteworthy though that it is recommended that you use an actual administrative Zendesk account rather than using just any agent with admin permissions.

Why?

When escalating a JIRA ticket via Zendesk, it will show up as created by whoever took care of the Zendesk integration. Using an account named Zendesk Admin or Zendesk Agent will definitely look better than Cedric Jacob. Especially if a lot of different Zendesk agents seem to act as Cedric Jacob.

The same thinking applies to the JIRA admin.

The second condition is also easy to grasp. Making sure that your JIRA account is in both of the mentioned groups (`jira-administrators` and `jira-software-users`) ensures that we possess the necessary permissions:

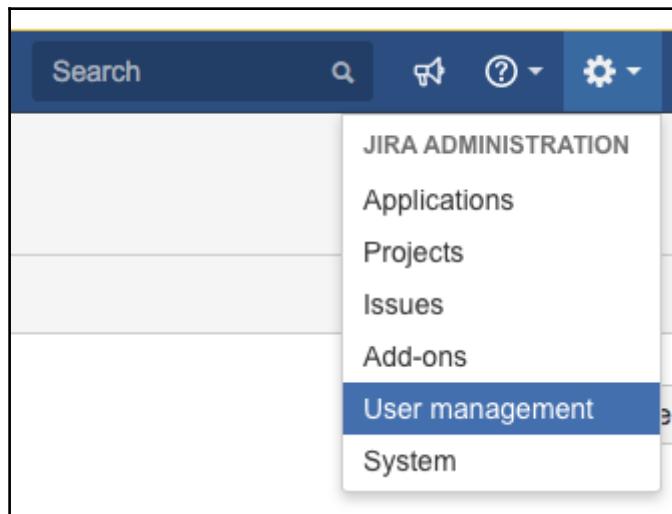
The `jira-administrators` group contains people who are JIRA system administrators. By default, this group is a member of the Administrators project role, and has the JIRA Administrators and the JIRA System Administrators global permissions.

– Atlassian Website

Users in the `jira-software-users` group have the Browse Users, Create Shared Filter, Bulk Change and Manage Group Filter Subscriptions global permissions.

– Atlassian Website

In order to do so, we need to access the **User management** page in JIRA:



The **User management** page allows us to find and edit users within the JIRA environment. If the user is not in the necessary groups, we can easily add them by clicking on **Edit**:

A screenshot of the JIRA User management page. The top navigation bar shows Add-ons, User management (selected), and System. The main area is titled "Users" and displays a table of users. A filter bar at the top left shows "cedric" in the "Filter users" field and "Any" in the "In group" dropdown. The table has columns: Full name, Username, Login details, Group name, Applications, Directory, and Actions. The "Group name" column for the user "Cedric Jacob" (username "cej") shows three groups: "jira-administrators", "jira-developers", and "jira-users". The "Actions" column for this user has a red box around the "Edit" button.



Note that groups may have been renamed. In my case, I had to choose `jira-users` instead of `jira-software-users`. After reviewing the group's permissions, I realized that this group had the permissions I was after.

Condition number three might not make sense straightaway. Let's elaborate on that one:

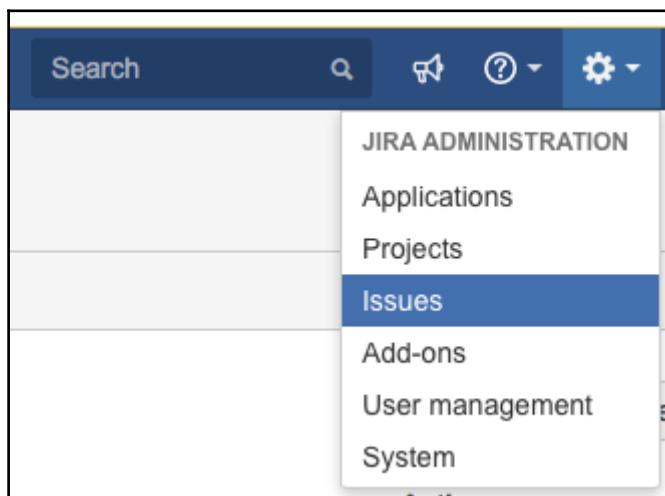
JIRA works similar to Zendesk when it comes to fields. But what does it mean when someone refers to a field on a specific screen?

Screens group all available fields (or a subset of all available fields) defined in JIRA and organize them for presentation to a user.

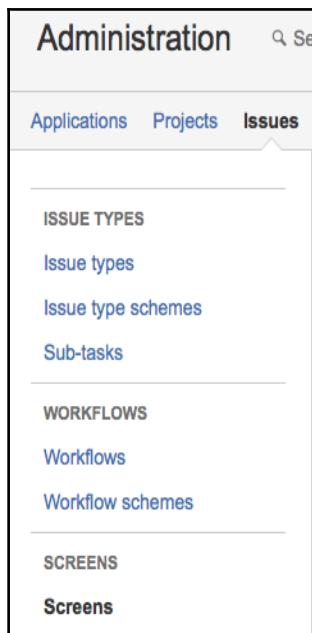
– Atlassian Documentation

In order to review and configure your screens in JIRA, follow these steps:

1. Access the administration menu by clicking on the wheel icon (upper-right corner):



2. Select **Issues** followed by **Screens**:

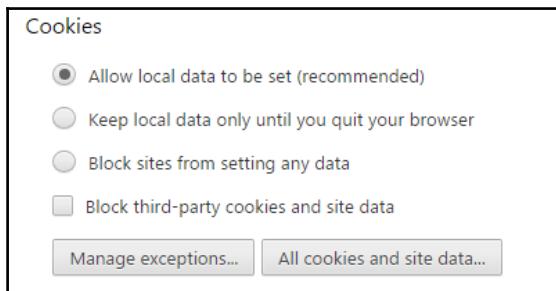


3. Click on **Configure** next to the screen. If the reporter field is not visible, add a new field by entering Reporter into the drop-down field at the bottom.



Condition number four is an easy one. If you are not sure whether the third-party cookies setting is enabled correctly in your browser, simply search online for the term `third-party cookies` in combination with your browser's name, such as Chrome. You will find a ton of different tutorials on how to change the settings within your particular browser.

On my Chrome browser, the settings in question look something like this:



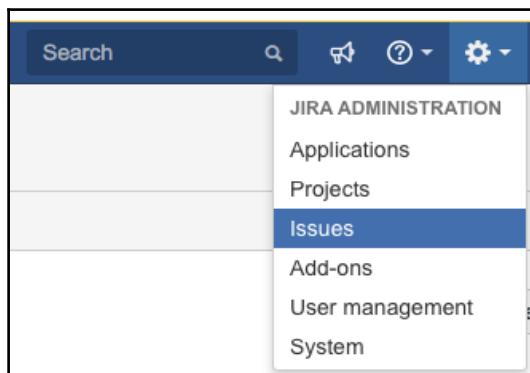
Note that if third-party cookies are disabled on purpose, add `jiraplug-in.zendesk.com` to the list of exceptions.

Last but not least, we need to make sure that the Atlassian add-ons possess the necessary permission to access our JIRA projects. We can do so by reviewing the Project Permissions page for the relevant projects in JIRA.

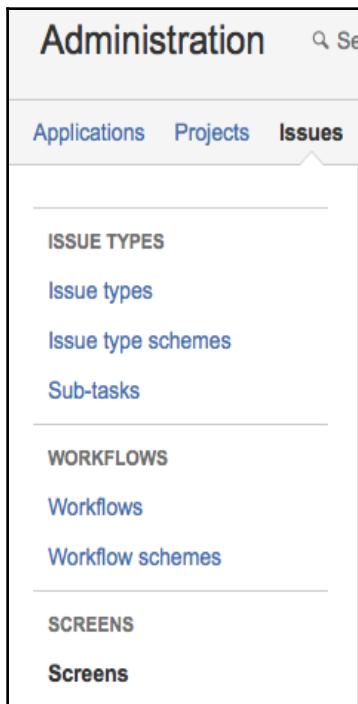
Project permissions are created within permission schemes, which are then assigned to specific projects by JIRA Administrators.

- Atlassian Documentation

So, in order for us to review the permissions, we will have to review our permission schemes by accessing the admin menu and choosing **Issues**:



Next, we locate and click on the **Screens** button on the left-hand side:



JIRA will display all our current permissions schemes and their associated projects. In order to access and change the permissions, simply click on **Permissions** under **Actions**:



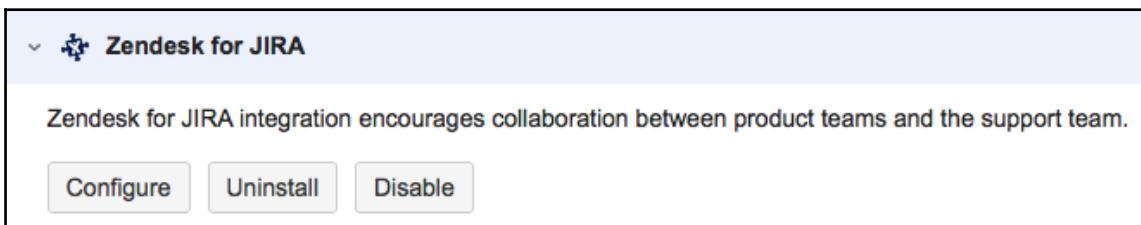
Note that if you are using JIRA Service Cloud, it is imperative to add a user called `add_on_zendesk_for_jira` to the `jira-users` group.

Installing the integration

Instead of installing a Zendesk app, we will do the whole setup within JIRA itself by installing an add-on and giving that add-on permission to access our Zendesk. However, do not worry, the corresponding Zendesk app will be installed automatically in the process.

Let's go through the setup:

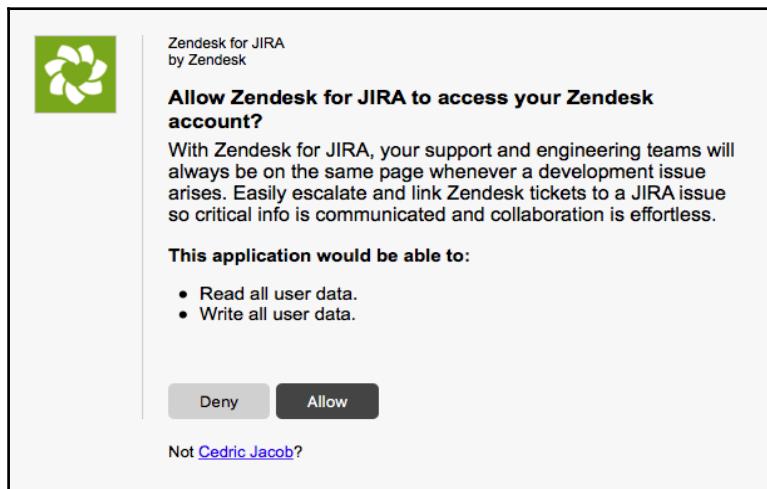
1. Access the administration menu by clicking on the wheel icon and selecting Add-ons.
2. Locate the Atlassian Marketplace header on the left-hand side and click on the **Find new add-ons** option right below it.
3. Enter Zendesk in the search bar.
4. Install the Zendesk for JIRA add-on by clicking on **install** located on the right-hand side of the search result.
5. After the installation is completed, repeat step 1.
6. Locate the Atlassian Marketplace header on the left-hand side and click on the **Manage add-ons** option right below it.
7. Locate the Zendesk add-on and open its context menu by clicking on it.
8. Click on **Configure**:



9. Enter your Zendesk subdomain, tick the ToS box below, and click on **Authenticate with Zendesk**:



10. JIRA will refer you to a Zendesk page (or log in to the page first). Grant the permissions and close this page:



- After returning to the JIRA page, we can choose the ticket fields we would like to link to JIRA:

The screenshot shows the Zendesk JIRA integration configuration interface. At the top, the Zendesk logo is displayed. Below it, the text "Your Zendesk: https://examplecomp.zendesk.com" and a "Disconnect your Zendesk" link. A section titled "Your JIRA base URL:" contains the value "https://jira.examplecomp.com". A descriptive text explains that users can select ticket fields to display on the right-hand side of linked JIRA issues. The "Your Zendesk ticket fields:" section lists various checkboxes for ticket attributes, with "Assignee", "Priority", "Group", "Type", "Status", and "Ticket Typ" checked. Other options like "Category" and "Kündigungsgrund" are unchecked. The "Other Zendesk information:" section has checkboxes for "Organization" and "Brand", both of which are unchecked. The "Other options:" section contains a checked checkbox for "Disable all public comments to Zendesk". At the bottom is a large green "Save my configuration" button.

- Click on **Save my configuration** followed by **Complete Set-up and Install App**.

That is it. We successfully integrated JIRA in our Zendesk environment. Alternatively, if you prefer, we integrated Zendesk in our JIRA setup.

When opening a ticket, you should see the following in the right app panel:



JIRA server specifics

When installing the add-on on JIRA Server, you may need to consider the following:

- Ensure that the right ports are open on your network. These depend on your SSL settings (usually: 80, 8080, 443).
- Allow Zendesk to connect to your JIRA Server via the list of specific public IP addresses. To find the updated list, simply search for `Zendesk Public IP addresses`.
- When installing the add-on in JIRA, you may have to change the “JIRA Base URL” (refer to the aforementioned Installing the Integration section, step 11). This URL must be publicly accessible.

Salesforce integration

If ExampleComp were using Salesforce, its integration surely would come with a lot of advantages.

Being able to view Salesforce data in Zendesk allows agents to make more informed decisions, but there is a lot more to it. Here are some of the other features:

- Sales reps can see Zendesk tickets in Salesforce
- Sales reps can send Zendesk tickets to Salesforce
- Sync organizations from Salesforce to Zendesk
- Sync users from Salesforce to Zendesk

Integration requirements

There are a few requirements that have to be met before we can go ahead with the integration:

- You must be at least on the Zendesk Team plan
- Your Salesforce organization must be at least on the Group edition
- If you are using Zendesk's host mapping, you will have to set up hosted SSL also
- You will need Salesforce and Zendesk admin access

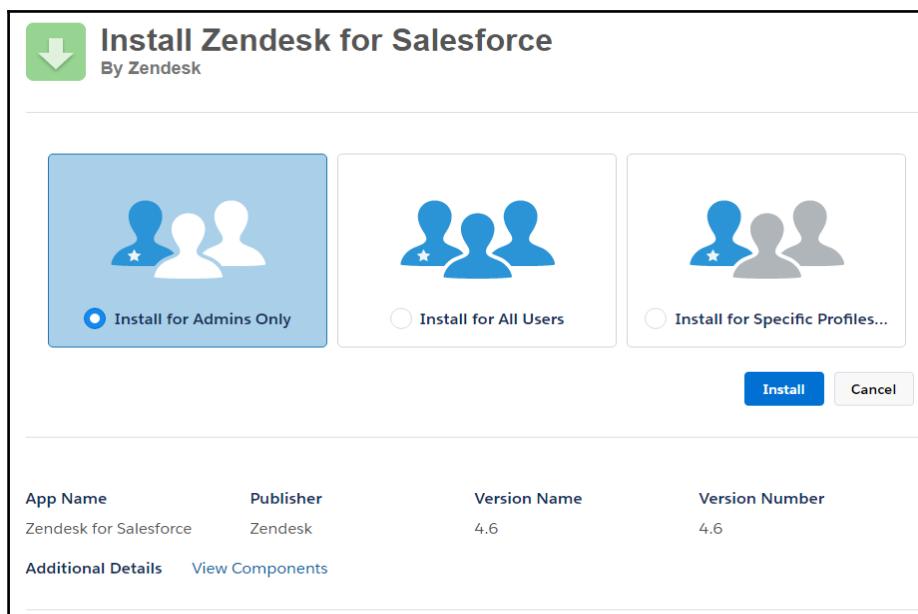
Installing the integration

First, we will have to install the *Zendesk for Salesforce* package. Here is the direct link to the package in question:

<https://login.salesforce.com/packaging/installPackage.apexp?p0=04t40000002eoC>

(You might have to log in to Salesforce.)

Select **Install for All Users** and click on **Install**:



Do not be alarmed. This might take a few moments. While the installation is in progress, you will see something like this:



Once the installation is complete, simply click on **Done**:

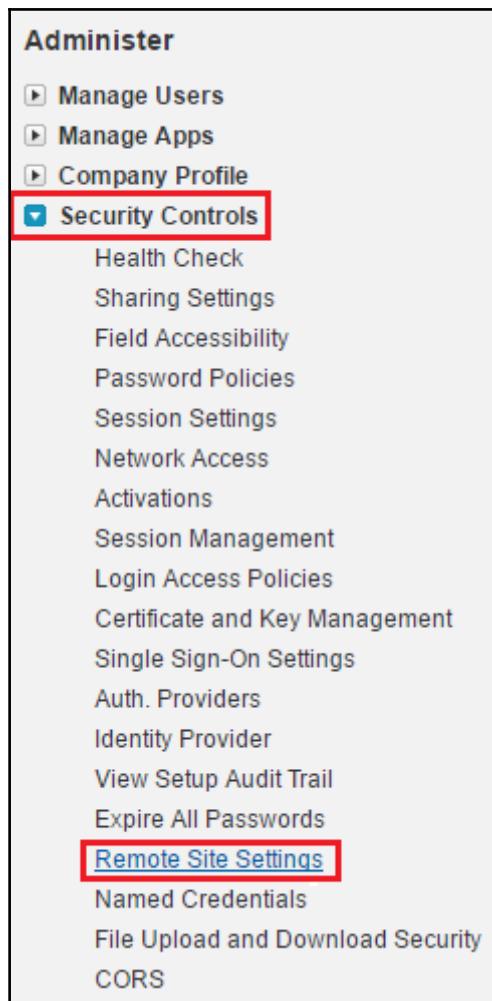


In order to allow back and forth communication between Zendesk and Salesforce, we will need to set up Zendesk as a remote site.

Navigate to the administration setup page by clicking on **Setup** next to your name:



1. Select **Security Controls**, followed by **Remote Site Settings** on the left-hand side:



2. Click on **New Remote Site**, as per the following screenshot:

All Remote Sites

Below is the list of Web addresses that your organization can invoke from salesforce.com. click New Remote Site.

View: All Remote Sites ▾ Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O

Remote Site Name ↑	Namespace Prefix	Remote Site URL	Active	Created By	Created Date
No records to display.					

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O

3. Enter **Remote Site Name** as well as **Remote Site URL** and click on **Save**:

Help for this Page

Remote Site Edit

Enter the URL for the remote site. All s-controls, JavaScript OnClick commands in custom buttons, Apex, and AJAX proxy calls can access this Web address from salesforce.com.

Remote Site Edit

Save Save & New Cancel

Remote Site Name: ExampleComp

Remote Site URL: ExampleComp.zendesk.com

Disable Protocol Security

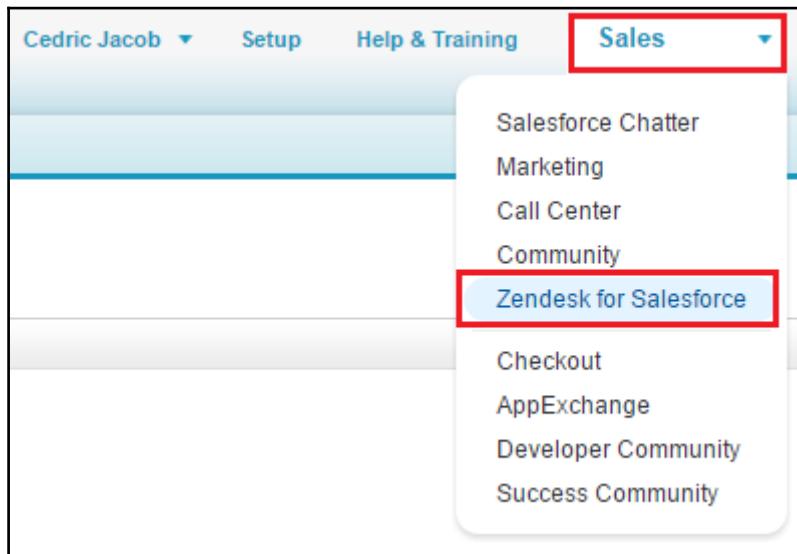
Description:

Active

Save Save & New Cancel

Next, we will need to authenticate to Zendesk. In order to do so, follow these steps:

1. Click on the button located in the top-right corner and select **Zendesk for Salesforce** from the context menu:



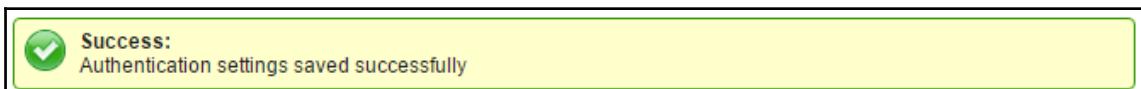
2. Click on **Zendesk** in the newly appeared navigation bar options:



3. Enter **Zendesk URL**, your Zendesk credentials, and click on **Authenticate**:

The screenshot shows the 'Global Settings' page for 'Zendesk for Salesforce'. At the top, there's a logo and the text 'Zendesk for Salesforce'. Below that, the heading 'Authentication - Global' is displayed. There are three input fields: 'Zendesk URL' with a question mark icon, 'User name' with a question mark icon, and 'Password'. Below these fields is a 'Authenticate' button.

If the provided credentials were correct, you should see the following:



Although we finished the initial setup, before we can reap the benefits of our labor, we will need to decide what features we want to take advantage of in the first place. Each will require some extra work.

Viewing Salesforce data in Zendesk

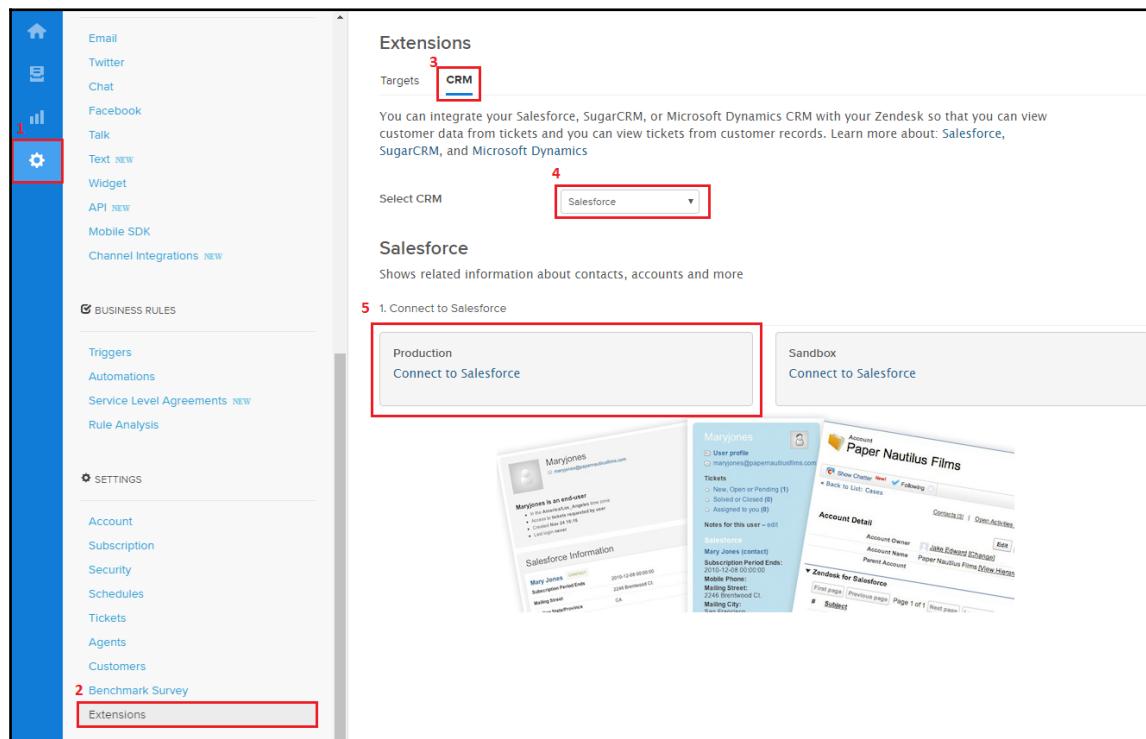
In order for our agents to make use of Salesforce data, we will need to facilitate some way of viewing this data within Zendesk itself.

First, we will need to install the Salesforce app from the Zendesk Marketplace.

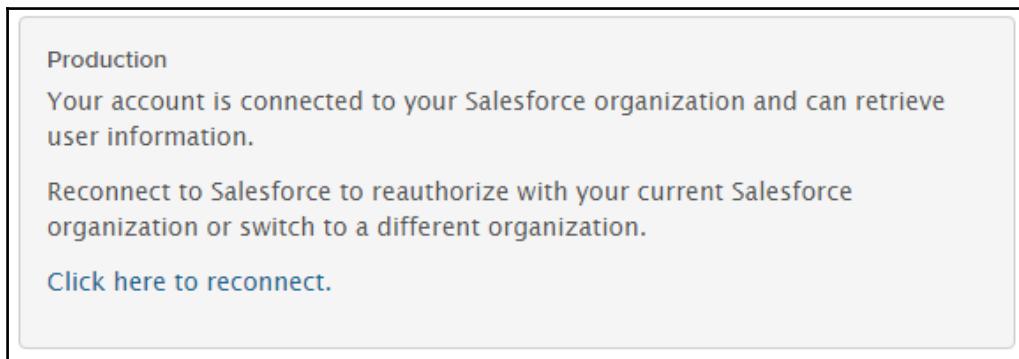
This is a very straightforward process, so go right ahead and take care of that (review *Zendesk app examples* again if needed).

Next, we will need to connect Zendesk to Salesforce. Simply follow these steps:

1. Click on the Admin icon (gear symbol) located at the bottom of Zendesk's sidebar.
2. Click on **Extensions** located under **Settings** within the admin menu.
3. Click on the **CRM** tab.
4. Select **Salesforce** in the **Select CRM** drop-down menu.
5. Click on **Connect to Salesforce**(you might have to log in to Salesforce):



If the connection was successful, you should see the following:



Next, we can take care of the app customization. Select **Customize new settings** from the drop down, as per the following screenshot:

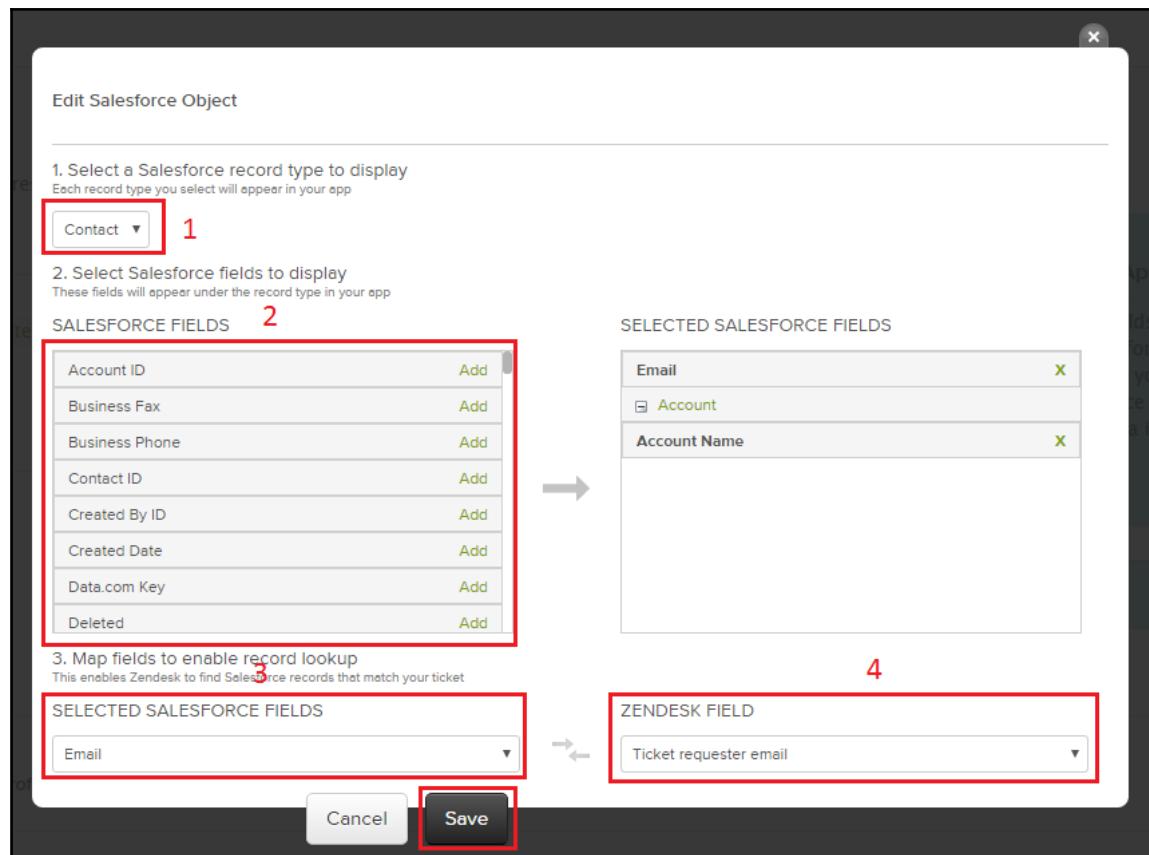
The screenshot shows a configuration interface for a Salesforce app. At the top left, the title "2. Customize your Salesforce app" is visible. Below it, a question "How do you want to configure your app?" is followed by a dropdown menu with three options: "Customize new settings" (highlighted with a red box), "Use existing settings", and "Customize new settings". A note below the dropdown says: "For integration features, you need to configure new settings. Your existing settings will be deprecated at some point in the future." To the right of the dropdown, there are "Add Object", "Edit", and "Remove" buttons. The "Edit" and "Remove" buttons are highlighted with red boxes. On the left side, there are sections for "Contact" and "Account". The "Contact" section includes "Full Name" and "Email Mapped to Zendesk: Ticket requester email". The "Account" section includes "Account Name". On the right side, a blue sidebar titled "Test your Salesforce App" contains instructions: "Enter test data in the fields below to test your Salesforce app. Make sure the data you enter is in your Salesforce organization so that data is returned." It also has a "Ticket requester email:" input field and a "Try It" button.

Zendesk will display a list of objects as well as the options to **Edit**, **Remove**, and **Add object**. Go ahead and remove any existing objects in order for us to start fresh.

Each object allows us to display a certain type of Salesforce data in our Zendesk app. So let's do exactly that and click on **Add Object**. Zendesk will prompt us to edit the Salesforce object first:

1. Select the **record** type, **Contact**, using the first drop down.

2. Add the Salesforce field, **Email**, by finding the **Email** field in the list and clicking on **Add**. Also add the **Account Name** field by finding the **Account Name** field in the list (located in the **Account** section) and clicking on **Add**.
3. In section 3, select **Email** in the first dropdown and **Ticket requester email** in the second.
4. Click on **Save**:

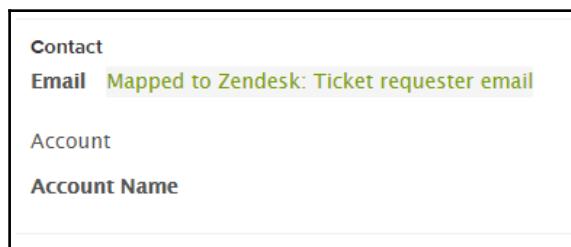


So what did we do exactly?

First, we chose to display the record type contact, which holds a customer's contact information within Salesforce. Then we went on to select two contact-related fields that we want to display within our app – the **Email** field and the **Account Name** field.

In order for Zendesk to find the data in Salesforce, we mapped the **Email** field with the **Ticket requester email** field in Zendesk.

Our list of object list should look something like this:

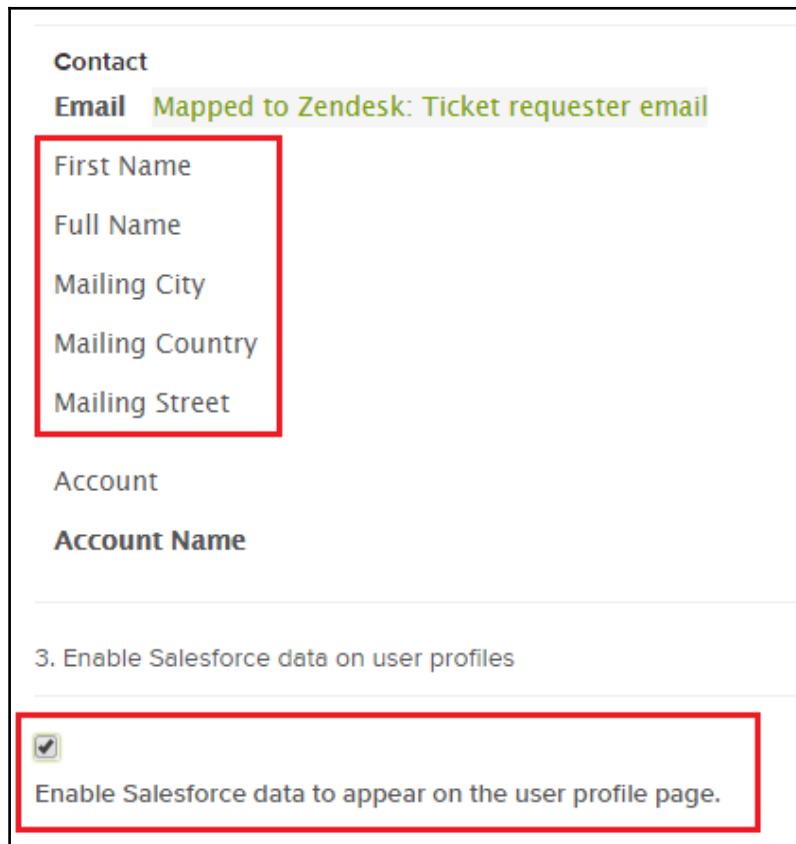


So Zendesk would use the requester's e-mail address to find the data in Salesforce and display both the **Email** address and the corresponding **Account Name** from Salesforce. Sounds good! But how about getting some more data from Salesforce?

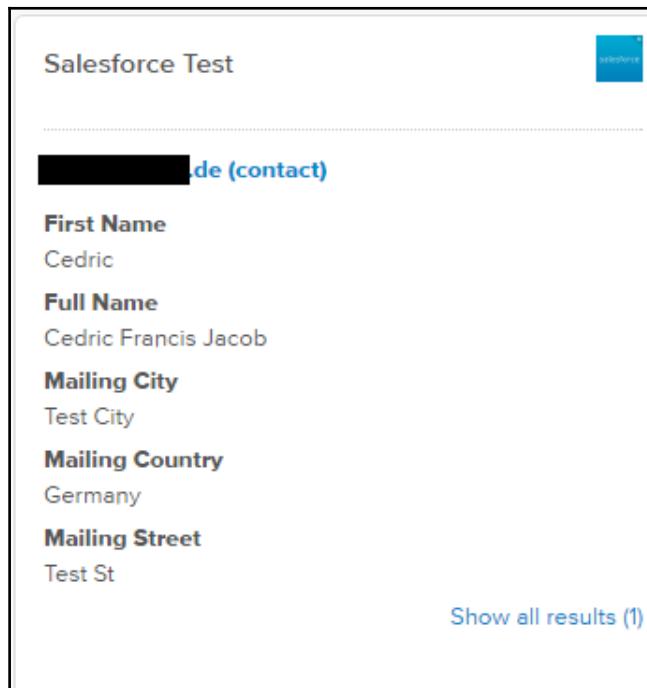
Let's edit our object again and add a few more fields:

- **Full Name**
- **Mailing City**
- **Mailing Country**
- **Mailing Street**

After doing so, make sure to tick the checkbox under our object list. This will enable the **Enable Salesforce data to appear on the user profile page** option:



Save the tab by clicking on **Save tab** at the bottom of the page before opening a ticket to test our app. Here we go, this is what our app should look like:



Granted for this to work, I had to create the contact in my Salesforce environment first.

Adding more features

One could surely write a small book about the features that come along with this integration. Luckily for us, we have covered the most important part of this undertaking and can make use of Zendesk's own documentation to set up additional desired features.

Here is the direct link to the necessary article:

<https://support.zendesk.com/hc/en-us/articles/203660216-Salesforce-Choosing-Zendesk-for-Salesforce-features>

An easier way might be doing an online search for *Salesforce: Choosing Zendesk for Salesforce features*, which should lead you to the same article.

Summary

In this chapter, you learned about Zendesk's integration and extension capabilities, the Marketplace, how to utilize its apps, and how to create a custom app using the ZAT. We also tackled two of the most complex integrations—JIRA and Salesforce.

In the next chapter, you will learn about advanced reporting options utilizing GoodData as well as Zendesk's new tool called *bime*.

7

Advanced Reporting and Insights via GoodData

When it comes to optimizing our Zendesk setup, not every customization can be derived from common sense and experience alone. Our decisions should be based on data. Data that describes our performance, namely our efficiency and customers' satisfaction.

Analyzing our performance helps us make important decisions when it comes to our staff, its training, our workflows, and overall Zendesk setup. We also gain some valuable insights about the current state of our business. Putting data into context, we can detect trends and create long term prognoses.

While Zendesk allows us to use its built-in reporting tools, we receive a second and more complex solution called Zendesk Insights utilizing GoodData for Zendesk. GoodData is a business intelligence tool developed by a third party and its sole purpose hinges on data.

In this chapter, we will review Zendesk's own reporting tools before moving on to the more complex GoodData solution.

This chapter covers the following topics:

- Creating and exporting reports in Zendesk
- Using Zendesk's Reporting Overview
- Using GoodData to create custom reports and metrics

Reports in Zendesk

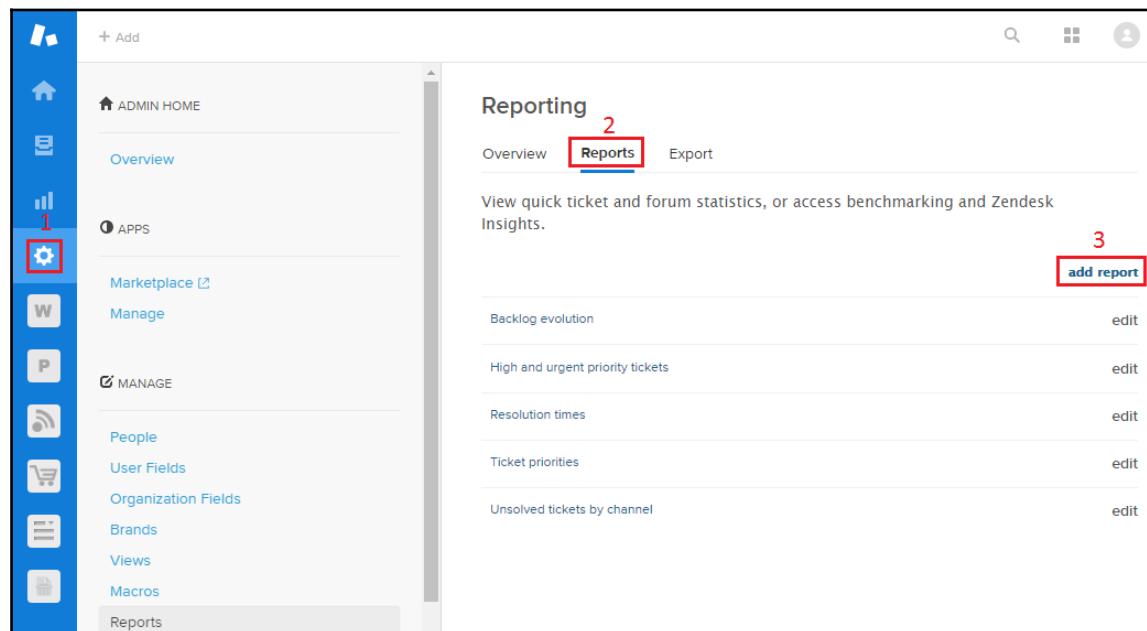
Zendesk allows us to quickly create reports within our support environment. While these reports are rather simple and we are more likely to use the more complex Insights later on, we should know how to create, review and export such reports if needed.

So let us go ahead and create our first report in Zendesk.

Creating and exporting reports in Zendesk

In order to create a report, please follow the given steps :

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Reports** located under **Manage** within the Admin-menu.
3. In order to create a new report, click on **add report** on the right side of the screen:



We will be presented with an empty report page waiting to be set up. We can divide the page into the following items:

- Report title
- Reporting period
- Data series

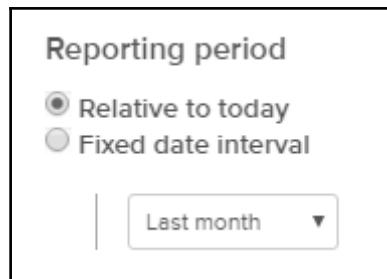
First we can choose a title for our report. We are going with **Created VS Resolved Problem Tickets (Last Month)**:

Report title

Created VS Resolved Problem Tickets (Last Month)

As the title of our report says, we are going to create a report that compares the amount of created problem tickets with the amount of resolved problem tickets within the same time frame of one month. So let us move ahead.

Next we will choose the reporting period, which in our case should be one month relative to today's date:



To complete our report, we will have to add the necessary data series. A data series contains the following three elements:

- The legend for the data series
- The actual data that we would like to display
- Optional condition(s)

In our case, we will need two **Data series** with one condition each. Let us have a look at what our first one should look like:

Data series

Created Problem Tickets Remove

Created tickets ▾

Ticket: Type ▾ **Is** ▾ **Problem** ▾ -

Add condition +

Our legend should read something like **Created Problem Tickets** and the data that we would like to display should be **Created tickets**.

Our condition is also self-explanatory: We want to display a Zendesk's Reporting Overview provides us with ticket where the **Ticket: Type** is set to **Problem**.

If this feels familiar, it is because the whole process of creating a report is very similar to creating Business Rules.

Our second data series is called **Resolved Problem Tickets**:

Resolved Problem Tickets Remove

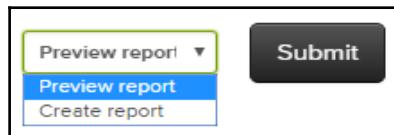
Resolved tickets ▾

Ticket: Type ▾ **Is** ▾ **Problem** ▾ -

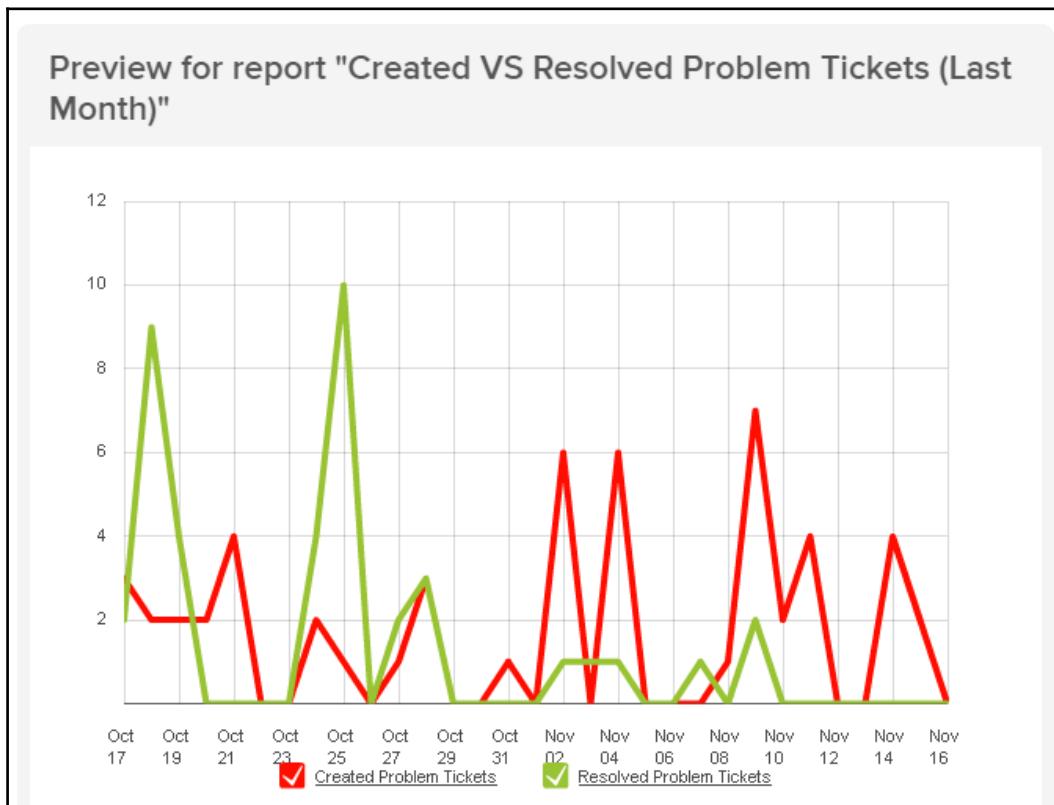
Add condition +

Instead of **Created tickets**, we want to display **Resolved tickets** instead. The **Ticket: Type** should be set to **Problem** as well.

Once we are happy with our settings, we can choose to preview the report and click on **Submit**:



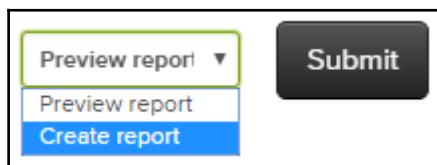
The preview consists of two parts. The first one is a graph:



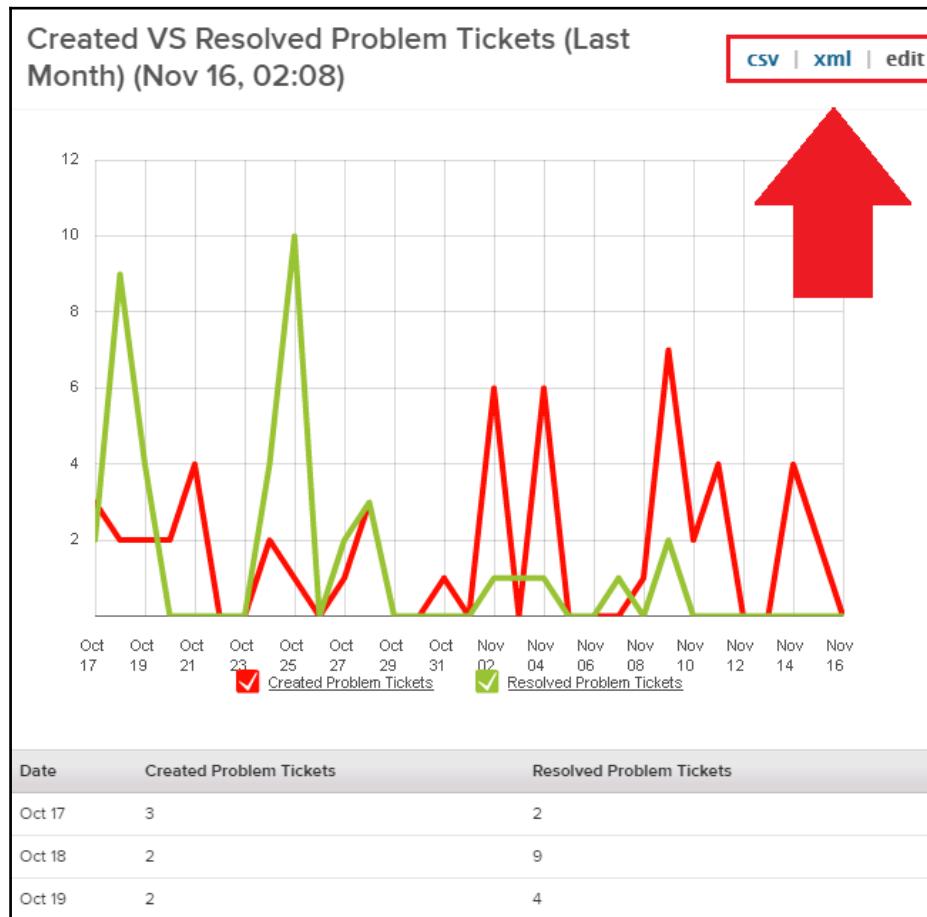
The second part is a data table:

Date	Created Problem Tickets	Resolved Problem Tickets
Oct 17	3	2
Oct 18	2	9
Oct 19	2	4
Oct 20	2	0
Oct 21	4	0
Oct 22	0	0
Oct 23	0	0
Oct 24	2	4
Oct 25	1	10
Oct 26	0	0
Oct 27	1	2
Oct 28	3	3
Oct 29	0	0
Oct 30	0	0
Oct 31	1	0
Nov 01	0	0
Nov 02	6	1

If we are happy with our report, we may decide to create it by changing the dropdown option and clicking on **Submit**:



Zendesk will display the updated list of reports and we can once more view our report by clicking on it:



Next to reviewing the graph and data table, we may choose to export our report as **csv**– or **xml**-file.

I would suggest creating a few more quick reports within Zendesk in order to develop a better understanding about the way these reports work. This understanding will become useful later on when working with more complex reports.

The Zendesk's reporting overview

Zendesk's Reporting Overview provides us with detailed performance reports related to our ticket volume, our agents and our Knowledge Base (Help Center).

In order to navigate to the Reporting Overview, simply click on the Reporting icon located in Zendesk's sidebar:



The Reporting View is subdivided into eight tabs:

- **Overview**
- **Leaderboard**
- **Knowledge Base**
- **Community**
- **Search**

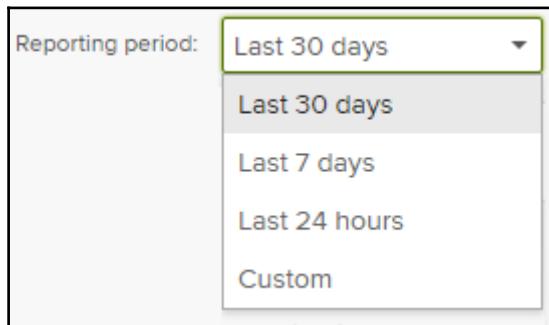
- Net Promoter ScoreSM
- Satisfaction
- Insights

The screenshot shows a navigation bar with tabs: Overview (underlined), Leaderboard, Knowledge Base, Community, Search, Net Promoter ScoreSM, Satisfaction, and Insights. Below the tabs, there is a "Reporting period" dropdown set to "Last 30 days".

Let us have a look at each one of the available tabs.

Overview

The overview offers a quick look at some of the most important key metrics. We can pick the desired time frame by choosing the **Reporting period** first:

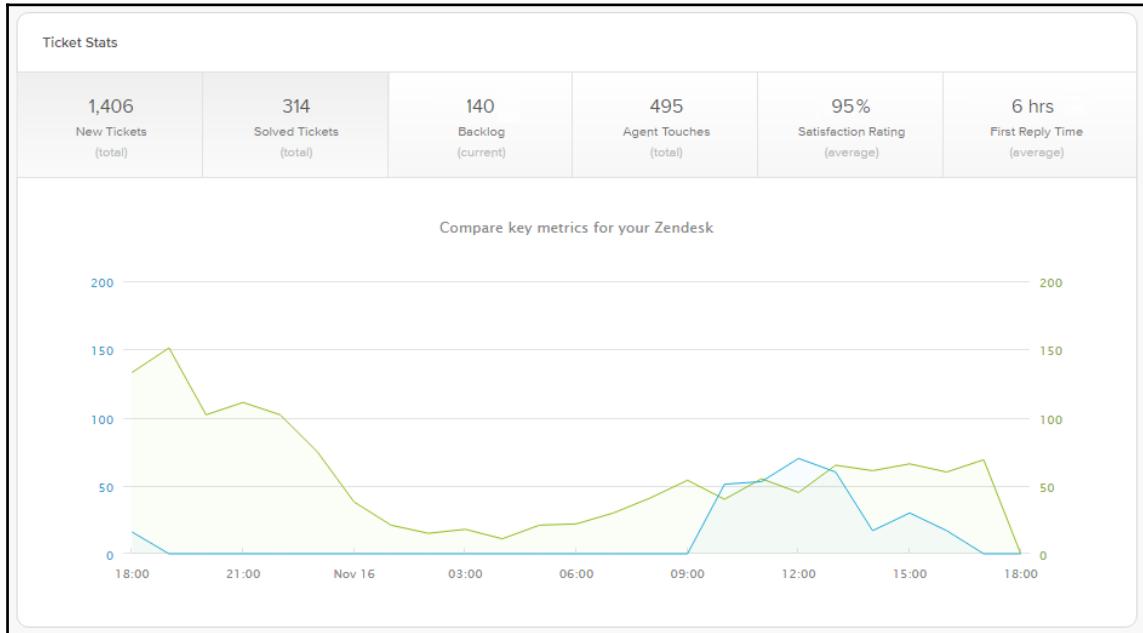


On one page, Zendesk presents us the following statistics:

- Tickets Stats
 - New Tickets (total)
 - Solved Tickets (total)
 - Backlog (current)
 - Agent Touches (total)
 - Satisfaction Rating (average)
 - First Reply Time (average)

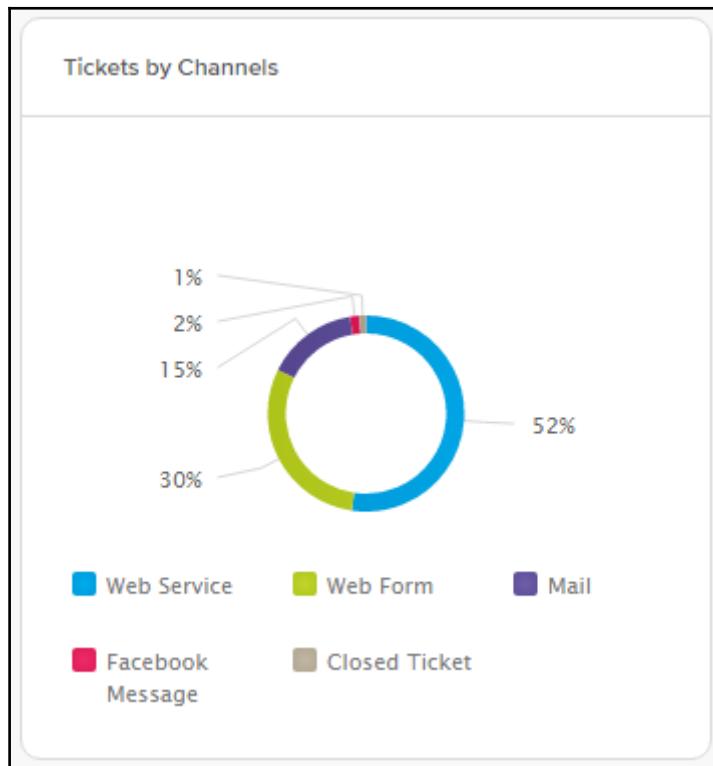
- **Tickets by Channels**
 - **Web Service**
 - **Web Form**
 - **Mail**
 - And more, depending on your open channels
- **Benchmark**
 - **Satisfaction Rating**
 - **Average First Reply Time**
 - **New Tickets**
 - **First Reply Time**
- **Help Center Content**
 - **By views**
 - **By net votes**
 - **By comments**
- **Top Searches**
 - **Total**
 - **Tickets created**
 - **With no results**
- **Top Agents**
 - **Tickets solved**
 - **Satisfaction**
 - **Touches**

The **Ticket Stats** are presented in total, average or current values as well as in a graph showing the development over time. Let us have a look at an example with a reporting period of 24 hours:



Zendesk allows us to pick one or two values to be displayed within our graph. We can change these values by clicking on the totals on top.

The **Tickets by Channels** statistics are displayed in a color-coded ring graph, where each color represents a different channel:



While each statistic is straightforward, easy-to-read and allows for a quick evaluation of our support, they surely cannot be utilized for in-depth analyzation. So let us move on to the other tabs.

Leaderboard

Leaderboards are a great tool to identify weak points. Again, Zendesk allows us to choose a **Reporting period**. Additionally we may choose to display the Leaderboard for **Groups** or individual **Agents**:

Reporting

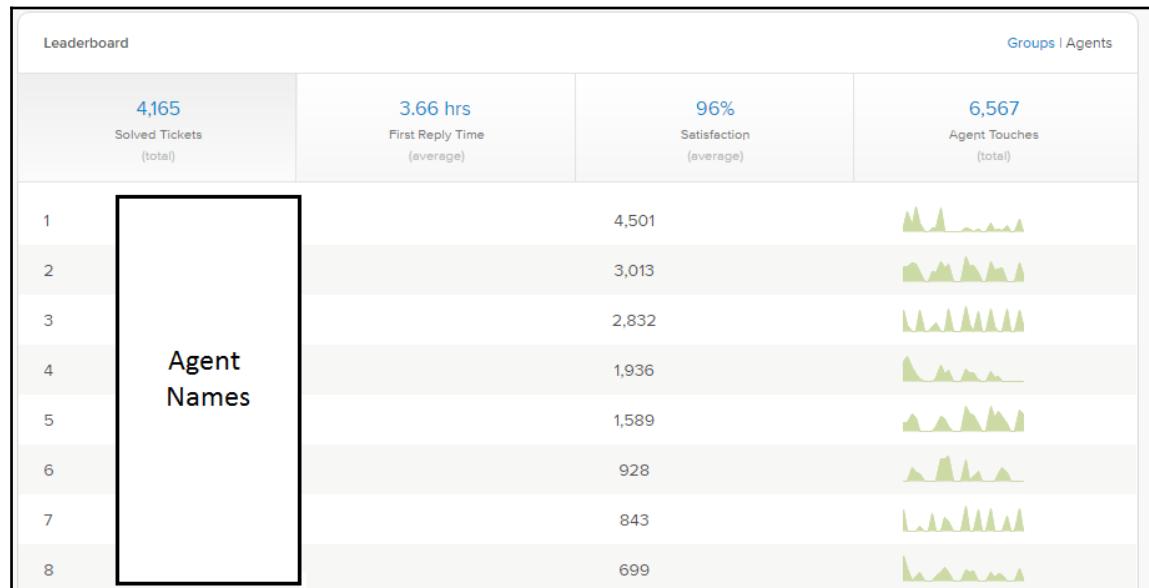
Overview Leaderboard Knowledge Base Community Search Net Promoter Score™ Satisfaction Insights Learn more

Reporting period: Last 30 days

Leaderboard Groups | Agents

Next up we can choose the key metric for our **Leaderboard** by clicking on the desired value:

- **Solved Tickets (total)**
- **First Reply Time (average)**
- **Satisfaction (average)**
- **Agent Touches (total)**



While the **Leaderboard** allows us to identify our strongest and weakest performers, in order to pin down the exact reasons behind these results, we will have to dig deeper.

For now, let us move to the next tab.

Knowledge Base

The **Knowledge Base** tab provides us with statistics related to our Knowledge Base / Help Center. Next to choosing a **Reporting period**, we may pick any amount of brands or channels we would like to review at one time. In our case we choose to review all brands and all channels:

A screenshot of a web-based reporting interface. At the top, there are two dropdown menus: "All brands" and "All channels", both with a downward arrow indicating they are filterable. Below these is a section for "Reporting period" with a "Custom" button. To the right of this are two date inputs: "Start: September 1, 2016" and "End: November 1, 2016". On the far right of this row is a blue "Update" button. The entire interface is enclosed in a thin black border.

The following metrics are displayed and can be viewed as a graph:

- **Articles (total)**
- **Views (total)**
- **Net Votes (total)**
- **Subscriptions (total)**
- **Comments (total)**

To put it in one sentence:

Zendesk allows us to review the number of articles created, the number of views for all our articles, the net result of votes on our articles (whether they were helpful or not), the number of new subscriptions as well as the number of new comments, all within the given time frame:



In addition, Zendesk compiles a list of our top articles depending on our metric choice and lists them below the graph:

Top articles by number of views	
	11761
	10120
	4036
List of article names and their authors.	3128
	2698
	2341

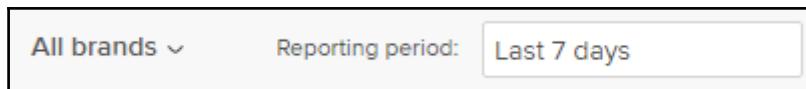
Reviewing the stats of our Help Center can lead to great findings and consequently to great improvements for our community.

We may for instance want to review all articles that lead to a bad Net Votes result.

Community

If we choose to activate the Help Center's community features, end users will have the capability to write posts, ask questions or provide feedback. This is where we can track the corresponding stats.

The **Community** tab is similar to our **Knowledge Base** tab, allowing us to filter by brands and **Reporting period**:



We can pick from the following metrics to display a graph:

- **Posts (total)**
- **Views (total)**
- **Net Votes (total)**
- **Subscriptions (total)**
- **Comments (total)**

Just like for the **Knowledge Base** tab, Zendesk will show a graph and compile a top list depending on the chosen metric.

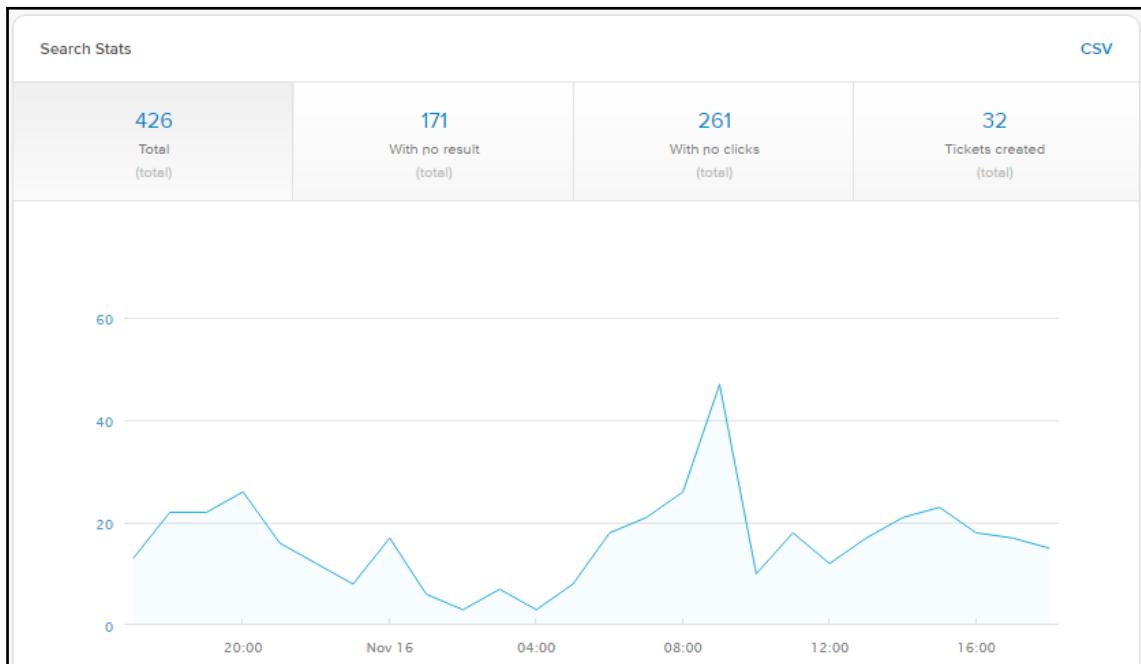
Search

The **Search** tab allows us to track our end user's search habits when visiting our Help Center.

We can filter by brands, channels and reporting period. All metrics are totals and directly linked to searches:

- **Total**
- **With no result**
- **With no clicks**
- **Tickets created**

The **Total** metric describes the number of searches, while each other metric shows a part of this total according to their title:



Following the graph, Zendesk compiles a list of **Search strings** along with the following values:

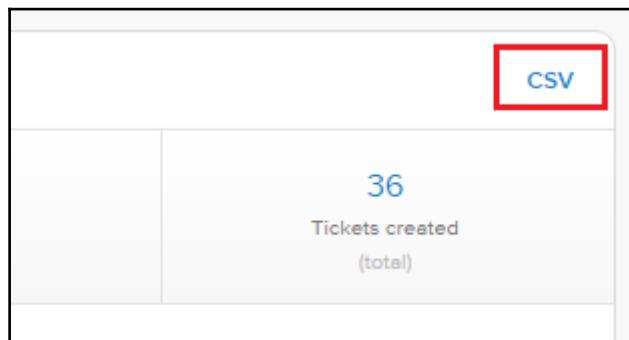
- **Total searches**
- **Avg number of results**
- **Click-through rate**
- **Tickets created**
- **Top clicked result**

The list looks something like this:

Search string	Total searches	Avg number of results	Click-through rate	Tickets created	Top clicked result
Search strings	235	11.6	91.1%	7	Direct links to top clicked
	203	6.7	89.2%	29	

This list is a great tool to identify search strings that may not lead to articles. It also helps to identify articles that may not be helpful enough to prevent the end user from creating a ticket.

I highly suggest exporting these lists via clicking on the **CSV** button in the right-upper corner:



Once exported we can use external tools such as Excel to analyze the data further.

Net Promoter Score

The Net Promoter Score tab only shows, once we have purchased the “Customer Lists & NPS Surveys” add-on.

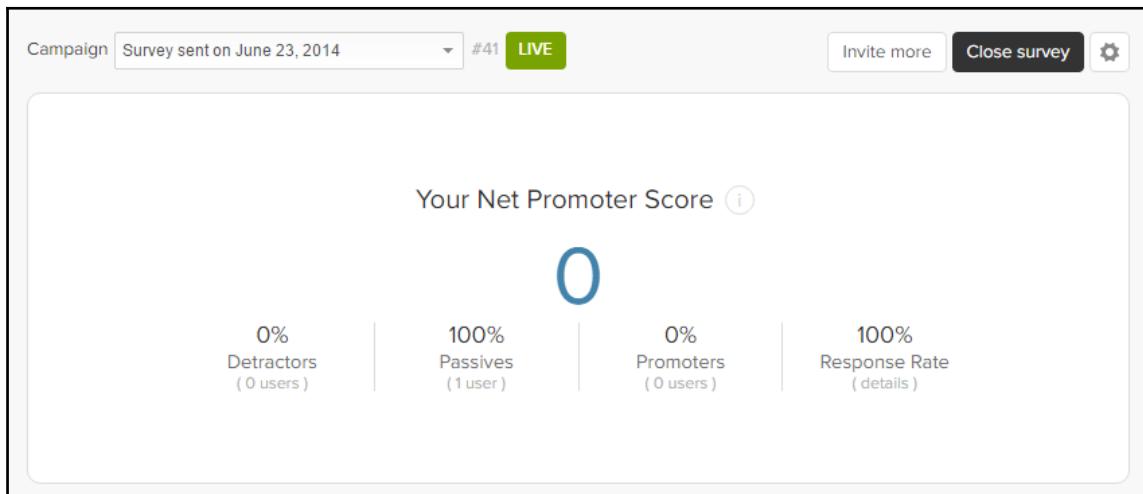
You can review available add-ons by visiting the following website: <https://www.zendesk.com/product/add-ons/>

What is the Net Promoter Score?

Segment your customers based on profile or support history. Create exportable lists or test your brand loyalty with Net Promoter Score surveys you can send to your lists right from within Zendesk.

—
Zendesk

Within the **Net Promoter Score** tab, Zendesk allows us to create such surveys, send them out to our customers and review the results:



Satisfaction

The **Satisfaction** tab displays statistics about our Customer's satisfaction. We can filter the data by choosing the reporting period.

The following values are being displayed:

- **Good ratings**
- **Bad ratings**
- **Surveys sent**
- **Response rate**



But where do these numbers come from?

We remember the automation “Request customer satisfaction rating” from Chapter 5, *Customizing Business Rules and Ticket Escalation*:

This automation is designed to send a notification to the requester asking him or her to rate their support experience, in Default Automations section, Chapter 5, Customizing Business Rules and Ticket Escalation.

And how do we get to the final satisfaction rating?

The satisfaction score is calculated according to the following formula:

$$\text{Satisfaction Score in \%} = \frac{\text{Good ratings in X amount of days}}{\text{Bad ratings in X amount of days}} \times 100$$

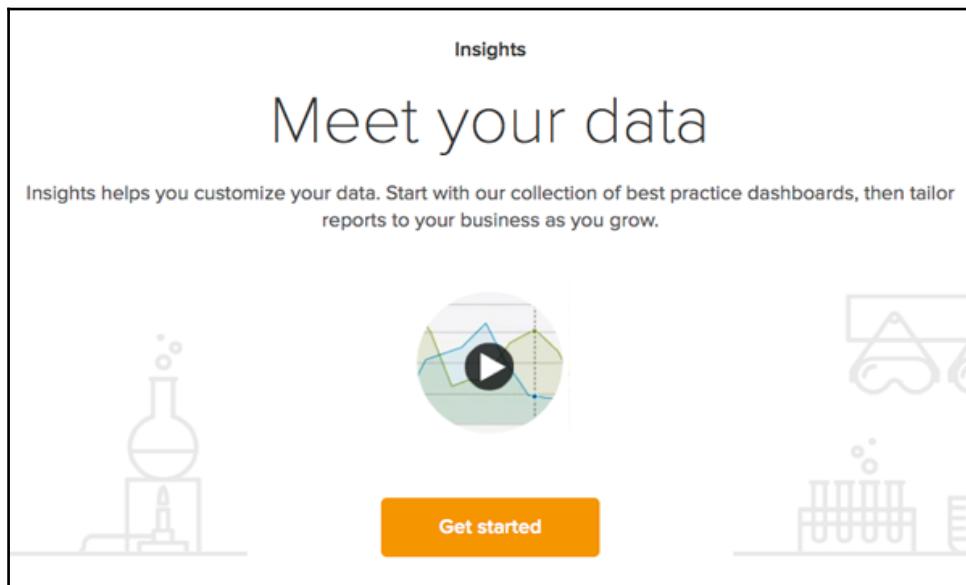
Zendesk also displays a list of all the ratings and their corresponding tickets:

Feedback ⓘ				Download CSV
Satisfaction: All	Comments: All			
Assignee	Ticket number	Satisfaction	Comment	
Agent Names	#488867	Good		
	#476685	Good		

The satisfaction score is a great indicator about how we are doing as a support team. If the numbers are bad, we should start investigating the cause.

Insights

The **Insights** tab is meant to display embedded reports from the external service called GoodData. Before we can start using this service, we will have to enable it by clicking on **Get started**:



Once we have enabled the service, a bunch of new statistics and graphs will start showing up within the **Insights** tab. That is because GoodData for Zendesk comes with a bunch of standard reports. However, it will become necessary to create your own reports at some stage. So let us dive into it.

Zendesk Insights/GoodData

Before we start creating our own reports, let us go through the overall structure and the corresponding terminology first.

Zendesk refers to Insights as a science lab for data. It allows us to examine our data more closely. Zendesk Insights is powered by GoodData, an external service. GoodData receives our Zendesk data and provides the functionality to create more complex reports, which are then embedded within our Zendesk environment.

When enabling Insights in Zendesk, we automatically receive access to a GoodData account. As this account is automatically linked with our Zendesk account, we do not have to worry about an initial setup.

Our GoodData project contains dashboards, which are divided into tabs. Each of the tabs is populated with reports.

For instance:

One of our prebuilt dashboards is the “Insights” dashboard. This dashboard contains tabs like these:

- **Overview**
- **Tickets**
- **Satisfaction**
- **Prediction**
- **Efficiency**
- **Agent Activity**
- **SLAs**

If we want to, we can create our own dashboards, add different tabs and populate them with pre-existing reports. We can also create our own reports entirely. We may also add filters that allow us to limit the scope of our reports. Filters are a great tool when it comes to trying to isolate important data correlations.

Let us have a look at **Insights** within our Zendesk environment:

The screenshot shows the GoodData Zendesk Insights interface. At the top, there is a navigation bar with tabs: Overview, Leaderboard, Knowledge Base, Community, Search, Satisfaction, and Insights. The Insights tab is selected and underlined in blue. To the right of the tabs is a "Learn more" link and the GoodData logo.

Below the navigation bar, there is a dropdown menu labeled "Insights - View Only" with the number "1" next to it. A red box highlights this dropdown.

Underneath the dropdown is a toolbar with several tabs: Overview, Tickets, Satisfaction, Prediction, Efficiency, Agent Activity, SLAs, Voice, and Learn More. The "Overview" tab is selected. To the right of the toolbar are buttons for "Default View", a lock icon, a refresh icon, and a print icon. The number "2" is placed near the toolbar area.

The main content area is titled "Overview". It includes a "Drill in supported" link. Below this are five filter dropdowns: TICKET GROUP (All), TICKET VIA (All), ORGANIZATION (All), CUSTOMER (All), and BRAND (All). A red box highlights this row of filters.

At the bottom of the main content area are three large rectangular boxes representing reports. The first report, highlighted with a red box and labeled "4", displays the text "Tickets created in the last 30 days" with a value of "11.9K" and a downward arrow indicating a decrease of "19%". The second report displays "Median first reply time" with a value of "3.0hrs" and a downward arrow indicating a decrease of "50%". The third report displays "Customer satisfaction" with a value of "-" and a downward arrow.

- This dropdown allows us to switch between dashboards
- These are the available tabs within the selected dashboard
- These are filters that allow us to narrow down our set of data
- This is one of the reports

I suggest clicking around a little. Switch between dashboards, check out the tabs and review the pre-existing reports.

You may find some of the reports helpful. In that case you could add them to your own dashboards later on.

While we can edit our project within Zendesk, in order to take full advantage of GoodData's functionality, we should switch to the GoodData environment. Any changes are applied to the embedded view within Zendesk.

We can switch to GoodData by clicking on **GoodData** in the top-right corner:



Zendesk will forward us to the GoodData environment, which, while looking similar, shows a few more options for us to play with:

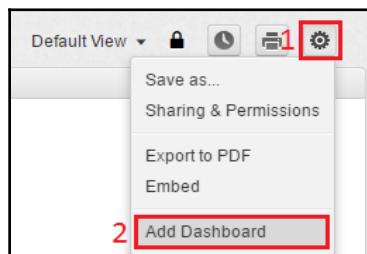
A screenshot of the GoodData Insights - View Only dashboard. The top navigation bar includes "GoodData", "GA Support", "Dashboards", "Reports", "Manage", and a user profile for "Cedric Jacob". The main title is "Insights - View Only". Below the title is a toolbar with "Default View", a lock icon, a clock icon, a printer icon, and a gear icon. The dashboard features three large cards: 1) "Tickets created in the last 30 days" showing "11.9K" with a downward arrow and "19%" in red. 2) "Median first reply time" showing "3.0hrs" with a downward arrow and "50%" in red. 3) "Customer satisfaction" showing a single dash "-". Each card has a "Drill in supported" link at the top right.

Now that we have a rough idea about GoodData, let us start by creating our own dashboard.

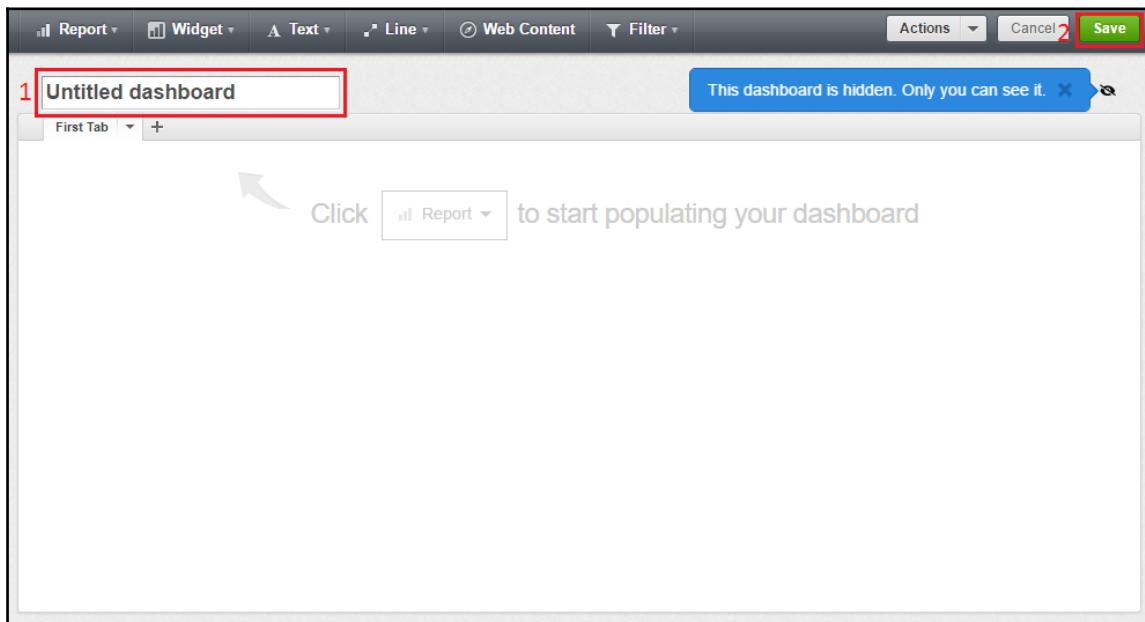
Creating a Dashboard and Tab

Follow the mentioned steps for creating a Dashboard and Tab:

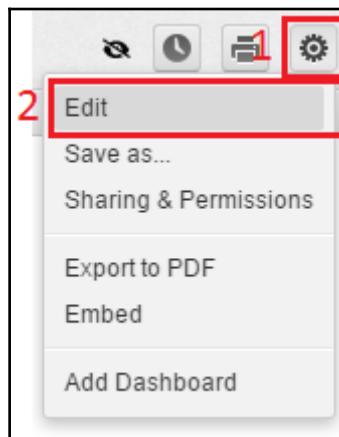
1. Creating a dashboard is simple. Simply click on the gear-icon in the top right corner followed by clicking on **Add Dashboard**:



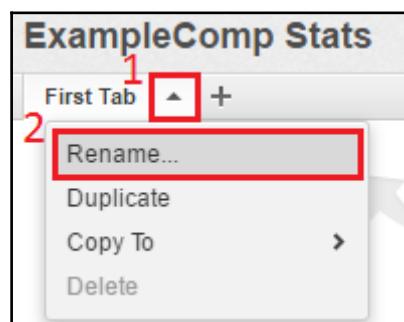
2. Next up, enter a name for our new dashboard and click on **Save**:



3. Before we can start by naming the first tab and adding reports, we will have to click on the gear-icon once more, followed by **Edit**:



4. Now we can click on the little arrow icon next to our first tab and choose **Rename...:**



In our case, let us call it `Ticket Stats`.

Great! We have created our own dashboard and named our first tab.

What can we do now?

GoodData gives us the option to add as many of the following items as we like:

- Report
- Widget
- Text
- Line
- Web Content
- Filter

We may not yet know what they are and what they are used for, but we will learn along the way. So let us go ahead and start by adding a report.

Adding reports

Finally, we are about to add our first report.

But what is a report in GoodData?

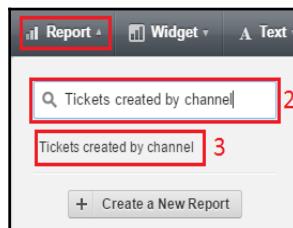
A report is a chart or table of metrics that has been designed to surface meaningful analytics into the GoodData Portal for stakeholders to review. A report contains one or more aggregation functions applied to numerical facts and optionally segmented by one or more attributes.

– GoodData

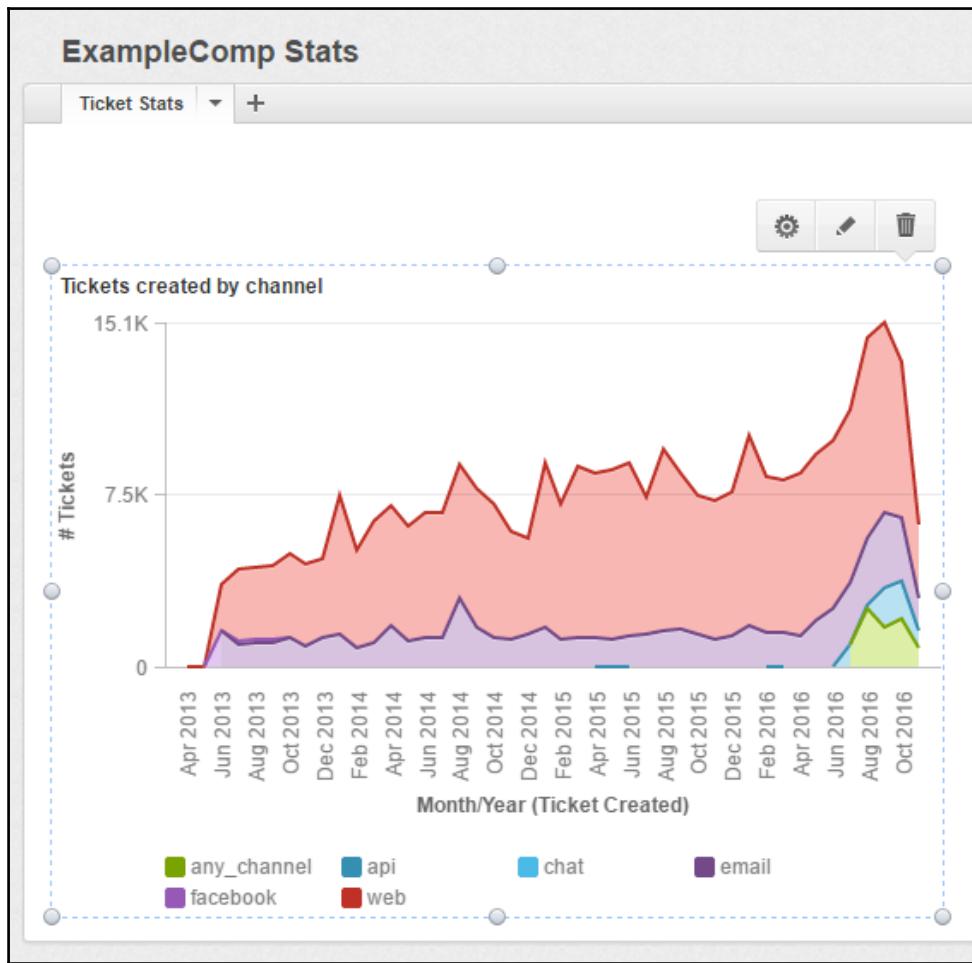
Documentation

In order to add a report, simply follow the given steps:

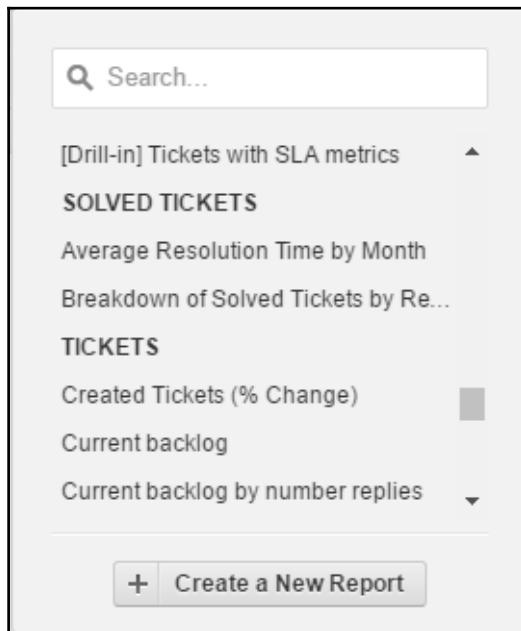
1. Click on **Report** in the top-left corner.
2. Enter Tickets created by channel in the search bar.
3. Click on the search result listed following the search bar:



The report will appear in our tab and we can commence by adjusting its size and location:



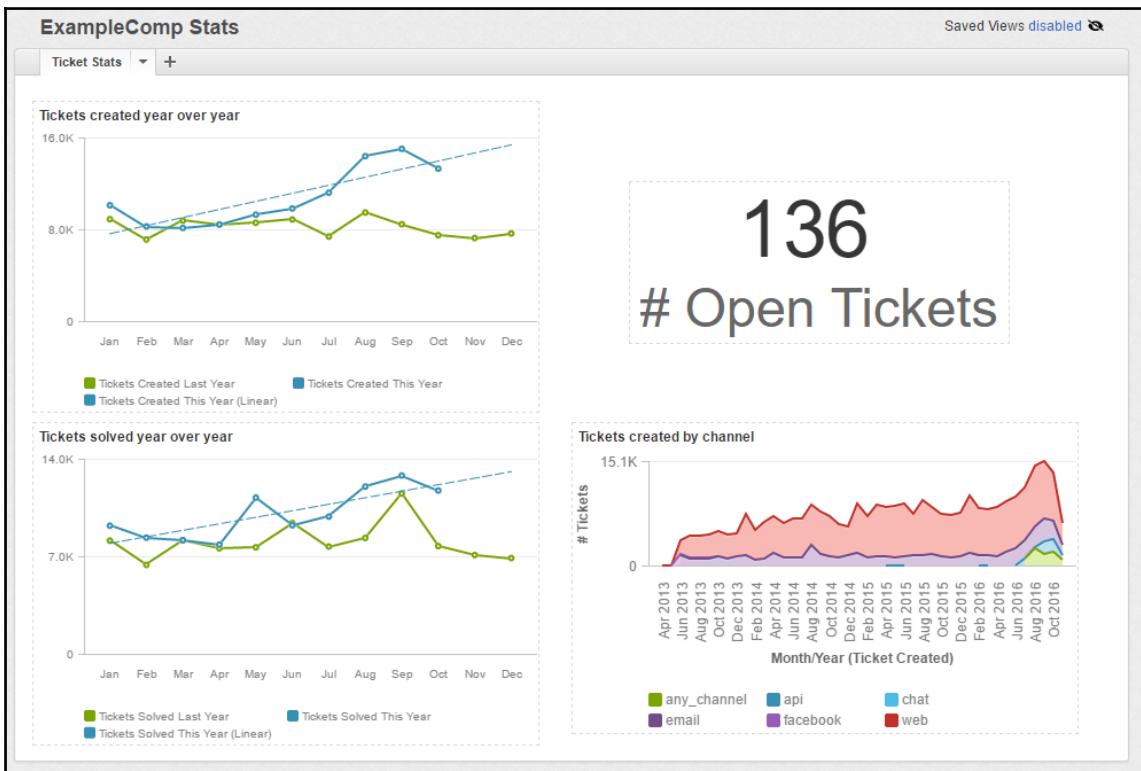
Go ahead and add a few more ticket-related reports. Instead of entering a search-string, you can scroll through all the existing reports, which are grouped by categories:



Some interesting reports might be the following:

- **Tickets created year over year**
- **Tickets solved year over year**
- **Current backlog**

After adding some additional reports, our tab should look something like this:



While adding pre-existing reports is easy, we probably want to come up with our own reports in order to really benefit from the newly gained insights.

Before we start creating custom reports, let us have a look at the other addable items in GoodData first.

Adding filters

Dashboard filters allow you to narrow the scope of one or more of the reports on the active dashboard tab. By selecting values in the filter, you can change the values displayed in the report without changing the report definitions.

– GoodData

Documentation

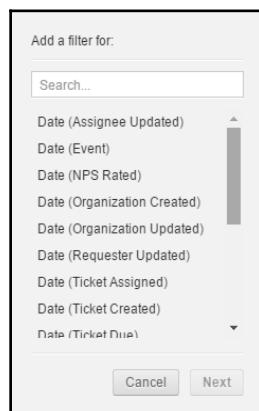
We have the option to filter by the following:

- **Attribute**
- **Date**
- **Group**

Let us start by adding a date filter. Being able to restrict the scope of our reports to certain time frames can be very helpful. In order to add a date filter, simply click on **Filter**, followed by **Date**:



GoodData will display another list:



We quickly realize, that there are many different date filters. To list a few:

- **Date (Ticket Created)**
- **Date (Ticket Assigned)**
- **Date (Ticket Solved)**
- **Date (Ticket Due)**

Looking at each filter, we can easily spot the reason for the great amount of different variations. A date filter only makes sense in context with an event.

But which one should we choose for our ticket tab?

Looking at our reports, we would only need a date filter for the event *ticket created* as the only report worth filtering would be “Tickets created by channel”.

The other reports are either set to display data year over year or the current state of our backlog. They surely do not need to be filtered by date.

So should we go ahead and create a filter using **Date (Ticket Created)**?

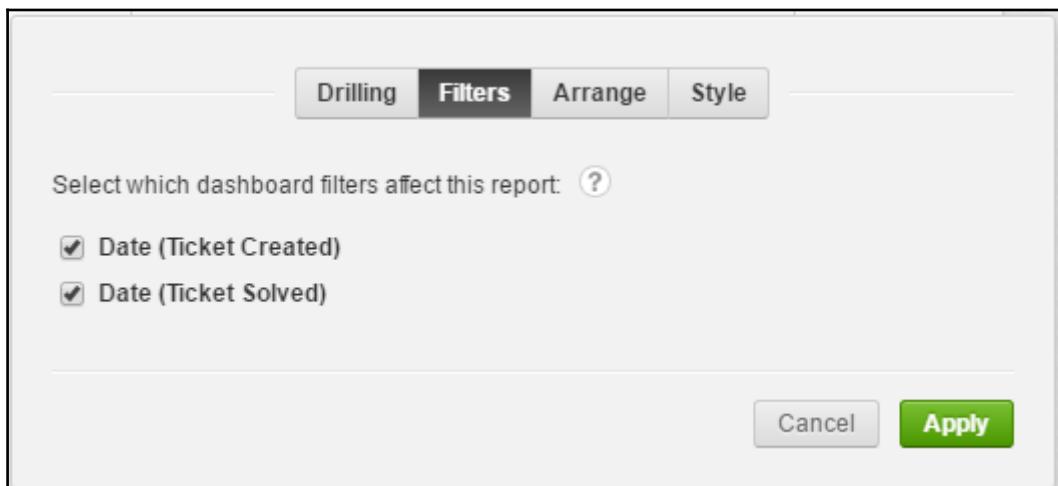
Well, we could.

But what if we had a lot more reports worth filtering, some using **Date (Ticket Created)**, some using **Date (Ticket Solved)**?

True! We probably may want to add more reports in the future. We could add both these filters, and then change the properties of each report to only be affected by the right date filter.

This can be achieved by following the given steps:

1. Clicking on the report.
2. Clicking on the gear icon that appears on the top-right corner of the report.
3. Clicking on **Filters**.
4. Choosing the filter, that should affect the report.
5. Clicking on **Apply**:



While possible, this practice would surely get out of hand. Just imagine having to set up 20 or 30 reports in one tab, having to add filters, to decide which filter should affect which report, and changing each setting accordingly.

Luckily for us, all pre-defined reports are set up in a way, that they allow us to use one specific date filter only: **Date (Timeline)**

But how does this work?

For now, let us say that the report itself holds the necessary information for the filter to know which date filter to apply. If no filter applies, the report will not be affected by the filter at all. We will learn how to do that later on when creating our own custom reports.

So let us go ahead, and click on the filter we would like to use (**Date (Timeline)**) followed by **Next**.

GoodData will ask us to select the default date range of our filter:

Select default date range:

Day Week Month Quarter Year

single day multiple days (continuous)

Select last: [7](#) [30](#) [60](#) [90](#) [180](#) [365](#) days

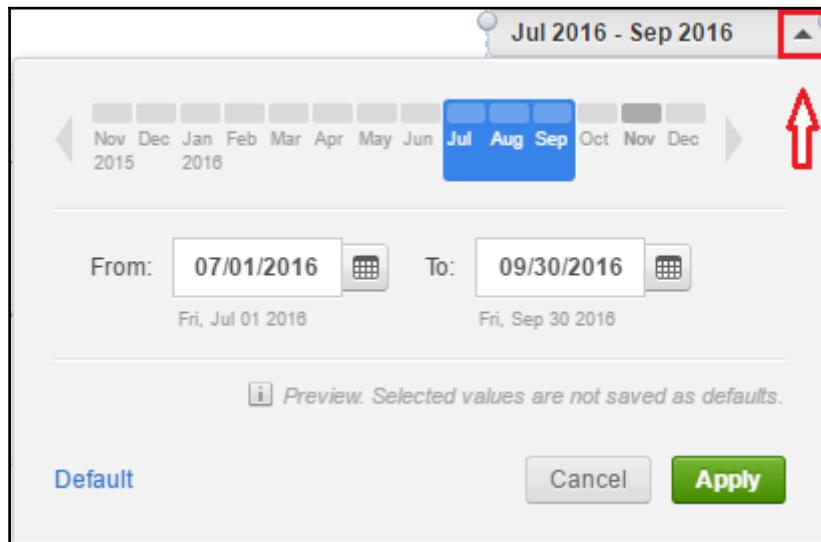
[► Advanced](#)

Preview: Oct 18 2016 - Nov 16 2016
Dates are based on current date in "Pacific Standard Time UTC -08:00" time zone.

Cancel

Let us pick **Month** as our report, the only one worth filtering right now, is using that measure of time. Confirm by clicking on **Add**.

After, in order to change the date-range and to apply the filter to our report, we will need to click on the arrow located on the right of our filter:

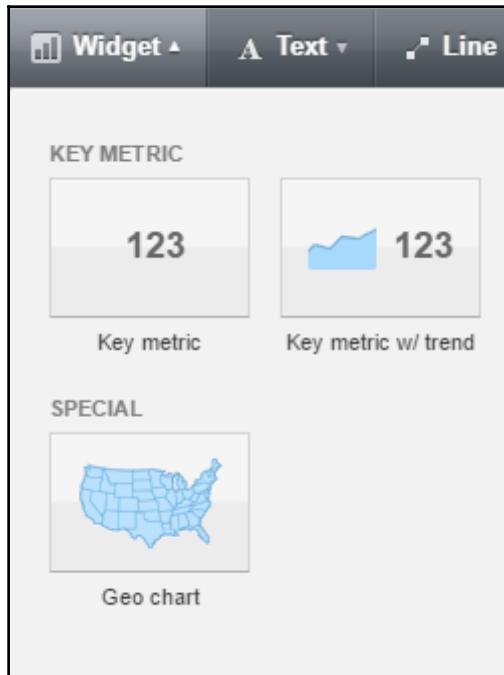


I would encourage you to experiment with some of the other filters. Try adding a group filter and different attribute filters. The concept remains the same.

Adding Widgets

Widgets are simplified reports. We can define a widget directly from the dashboard by selecting a metric and a time-period. A special type of widget, the Geo chart widget, contains a heat map that allows us to visualize data broken down by region across an actual map.

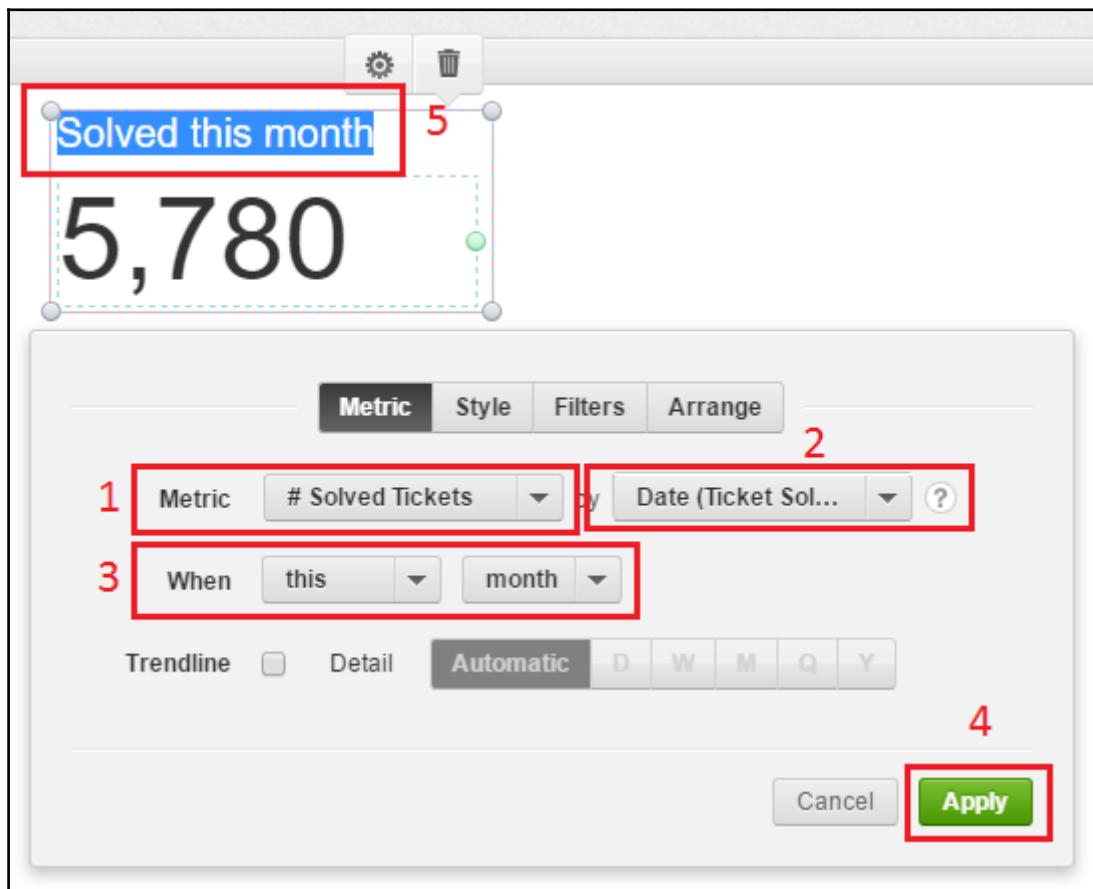
Let us go ahead and add a widget by clicking on **Widget**, followed by **Key metric**:



GoodData will ask us to set up and name the widget. In order to do so, please follow the given steps:

1. Pick the Metric **#Solved Tickets**.
2. Choose **Date (Ticket solved)**.
3. Pick the time period **this month**.

4. Click on **Apply**.
5. Name the widget:



Widgets are great to display simple key metrics that we might be interested in.

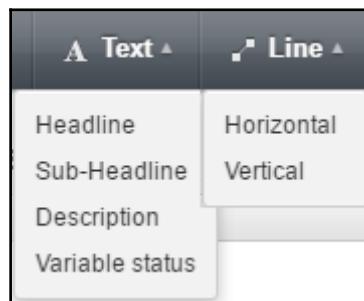
I suggest you try adding a few more key metrics. Here are some suggestions:

- Tickets created last week
- Tickets created this week
- Tickets solved last week
- Tickets solved this week

Adding some text and lines

Adding texts and lines can really help to make our dashboard more presentable and easy-to-read. You may add as many text and line elements as you wish.

Simply click on either **Text** or **Line** and choose the item you would like to add:



Try adjusting your tab to look something like this by deleting the unnecessary reports, moving elements around and adding the text and line elements needed:

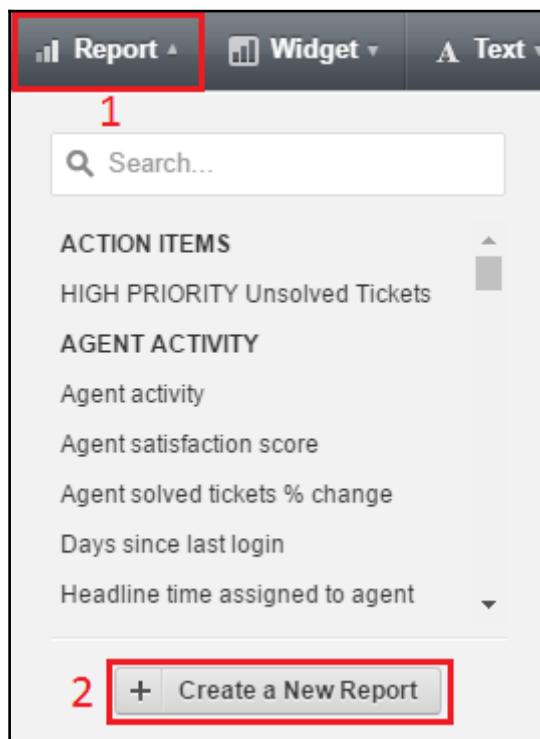
In order to save and review the final dashboard, simply click on **Save** in the right-upper corner.

What do we have in our tab now?

Some text, some lines, four key metrics and a date (timeline) filter, waiting for custom reports to be created and added to our dashboard's first tab.

Creating custom reports

In order to create a custom report, first click on **Report** in the left-upper corner, followed by **Create a New Report**:



Zendesk will present us with an empty report, ready to be set up:

The screenshot shows the GoodData reporting interface. At the top, it says "Untitled report" and "Click here to add description". Below that are three buttons: "What", "How", and "Filter". A large text box below the buttons says "Build a report by asking What and How of your data...". To the left is a grid diagram with two columns and two rows. The top-right cell contains the word "What" and the bottom-left cell contains the word "How". Lines connect the "What" and "How" labels to their respective definitions on the right. The "What" definition is "Metrics (e.g., Revenue) What numbers you would like to see." and the "How" definition is "Attributes (e.g., Year) How you would like to break your numbers down."

Creating GoodData reports is as simple as answering two questions:

- **What** do we want to display?
- **How** do we want to display it?

Let us go ahead and create a report that displays how many tickets have been created per week. We can start by picking a name for our report:



Next, we decide what we would like to display. In our case, we want to display the amount of tickets:

1. Click on **What**.
2. Enter # tickets in the search bar.
3. Tick the box next to # Tickets:

The screenshot shows the GoodData reporting interface in the 'What' configuration step. The top navigation bar includes 'What (1)', 'How', and 'Filter' buttons. Below this, a 'View by Folders' dropdown is set to 'Folders'. The main area is divided into 'Metrics' and 'Detail' sections. On the left, a tree view lists various metrics categories: Backlog, Business Hours, Calls, NPS® Surveys, Satisfaction Prediction, Satisfaction Surveys, System, Ticket Comments, Ticket Events, Ticket SLAs, Ticket Times, Tickets, and Tickets (1). The 'Metrics' section contains a search bar with '# ticket' and a list of metrics with checkboxes. The '# Tickets' checkbox is checked and highlighted with a red box and the number '3'. Other listed metrics include: # Ticket Events, # Ticket Updates, # Tickets Created, # Tickets Deleted, # Tickets Reopened, # Tickets Solved, and [GeoChart] # Tickets Created. At the bottom right are 'Cancel' and 'Done' buttons.

Now we need to decide how we want to display our chosen metric. We want to display our tickets divided by the week they were created:

1. Click on **How**.
2. Enter **Week** in the search bar, find the attribute **Week (Mon-Sun)/Year (Ticket Created)** and tick the box next to it:

The screenshot shows the 'How' configuration screen. At the top, there are tabs for 'What (1)', 'How (1)' (which is highlighted with a yellow box and a red number 1), and 'Filter'. Below these are sections for 'View by' (Folders) and 'Attributes'. In the 'Attributes' section, a search bar contains the text 'Week'. A dropdown menu lists various attributes, with the option 'Week (Mon-Sun)/Year (Ticket Created)' checked. Other listed attributes include various Day of Week and Day of Month options. At the bottom right are 'Cancel' and 'Done' buttons.



Be careful: Make sure that our attribute contains the string “/Year”. This assures that we get the # of tickets, not only restricted to a particular week, but a particular year as well. Otherwise we end up with the # of tickets for a particular week for all years combined.

Last but not least, we can choose a filter. We remember:

"For now, let us say that the report itself holds the necessary information for the filter to know what date filter to apply." in section Adding Filters of this chapter.

We want to make sure that our filter only holds the information regarding the type of filter that we want to use. Picking the time period should be up to the viewer of the dashboard by utilizing the **Date (Timeline)** filter there.

So let us go ahead and take care of that:

1. Click on **Filter**.
2. As we want to filter by a date range, click on **Numeric Range Filter**:

The screenshot shows a user interface for filter selection. At the top, there are three tabs: "What (1)", "How (1)", and a yellow-highlighted "Filter" tab. A red number "1" is placed next to the "Filter" tab. Below it, a callout bubble also contains the word "Filter".

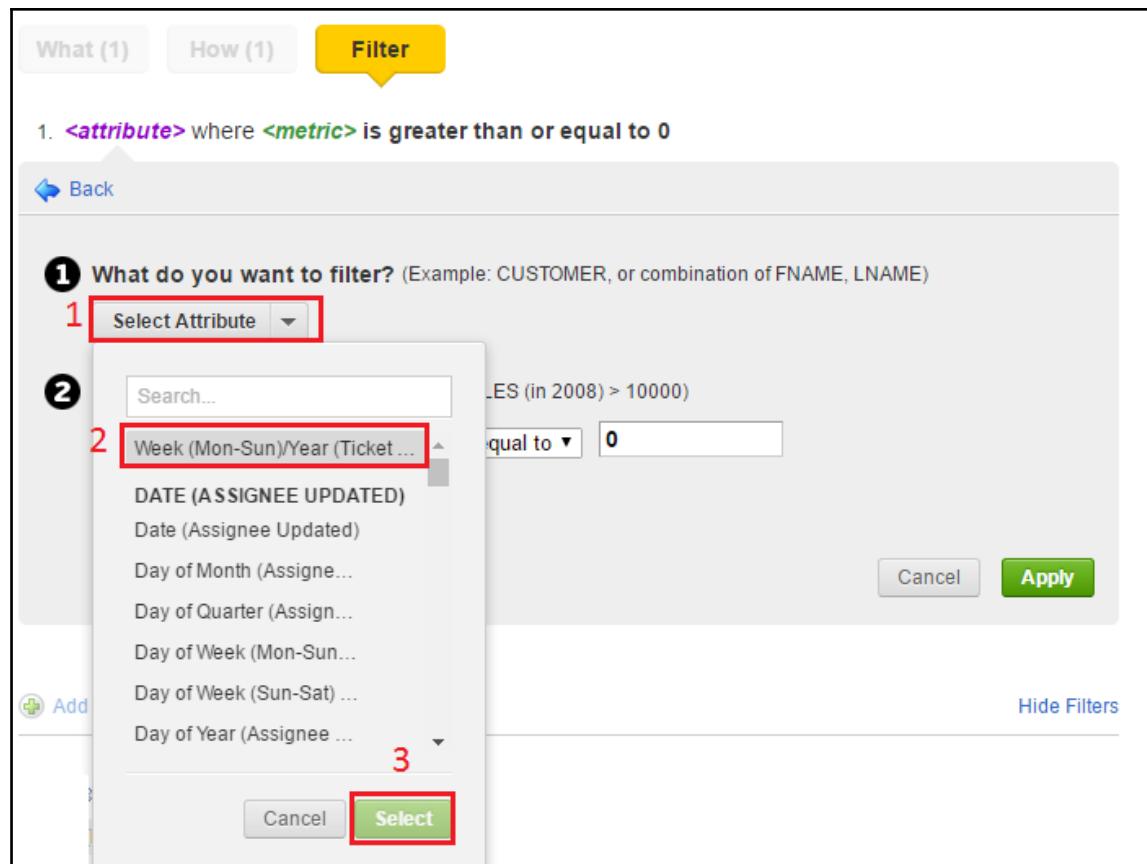
Under the "Filter" tab, the text "Select one of the filter types below:" is displayed. There are four filter options listed:

- Select from a List of Values (including date ranges)**: Description: Pick values from a list. (Example: Year is 2006, 2007 or "Last 4 Qtrs")
- Ranking Filter**: Description: Rank your results. (Example: Top 10 Stores by Sales, Bottom 20 Outlets by Expenses).
- Numeric Range Filter**: Description: Specify a range for your numbers. (Example: Stores where Sales is less than 10000 in 2008). This option is highlighted with a red border and a red number "2" is placed to its right.
- Variable Filter**: Description: Look for a variable. (Variables display dynamic data based on values stored in a specific user's profile.)

At the bottom left is a "Add Filter" button with a plus sign icon. At the bottom right is a "Hide Filters" link.

First GoodData will ask us to provide the attribute that we want to filter:

1. Click on **Select Attribute**.
2. Choose **Week (Mon-Sun)/Year (Ticket Created)** located at the top.
3. Click on **Select**:



Next we will provide the metric and the range of the filter:

1. Click on **Select Metric**.
2. Choose **_Filter Ticket Created Date**.
3. Click on **Select**:

The screenshot shows the GoodData Filter dialog with the following steps highlighted:

1. What do you want to filter? (Example: CUSTOMER, or combination of FNAME, LNAME)
Week (Mon-Sun)/Year (Ticket Created)
2. Select metric and range (Example: SALES (in 2008) > 10000)
Select Metric: is greater than 0
3. _filter
_Filter Event Date
_Filter NPS® Rated Date
_Filter Org Created Date
_Filter Ticket Assignee ...
_Filter Ticket Created Date
_Filter Ticket Solved Date
_Filter Updater
_Filter User Created Date
4. Select

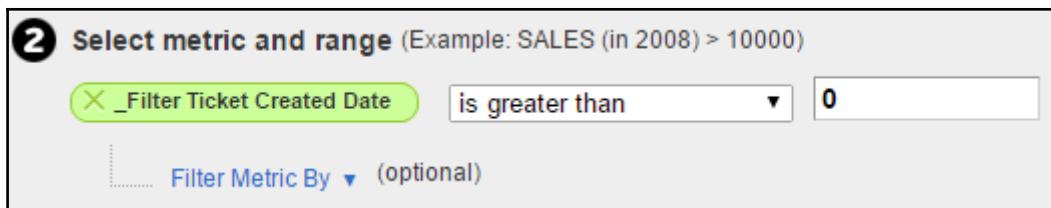
Buttons visible in the dialog include Back, Add Another Attribute (Optional), Cancel, Apply, and Hide Filters.

By picking **_Filter Ticket Created Date**, we are telling GoodData that we are relying on an external **Date (timeline)** filter and that our report should be filtered by the date when the ticket was created.

Also make sure to pick **is greater than 0**, which requires the date range to be at least one day:

2 Select metric and range (Example: SALES (in 2008) > 10000)

is greater than
Filter Metric By ▾ (optional)

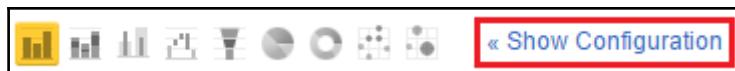


In order to preview the report, click on **Apply**. A new set of options will appear in the form of icons allowing us to pick the way our report should be visualized.

Our report is currently displayed as a table. In order to change it to a bar chart, click on the corresponding icon:

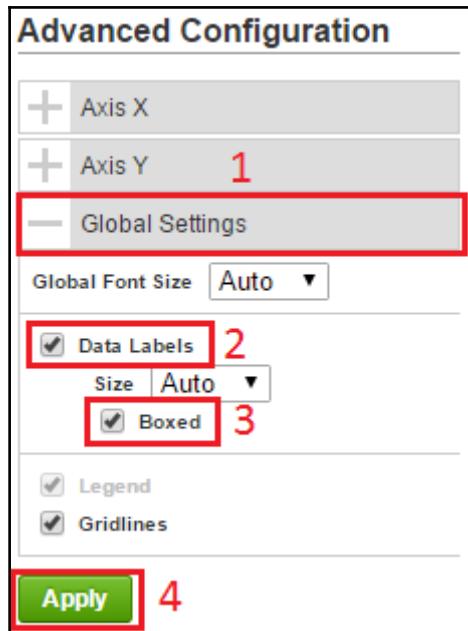


Next, we need to bring up the bar chart configuration options by clicking on **Show Configuration**:



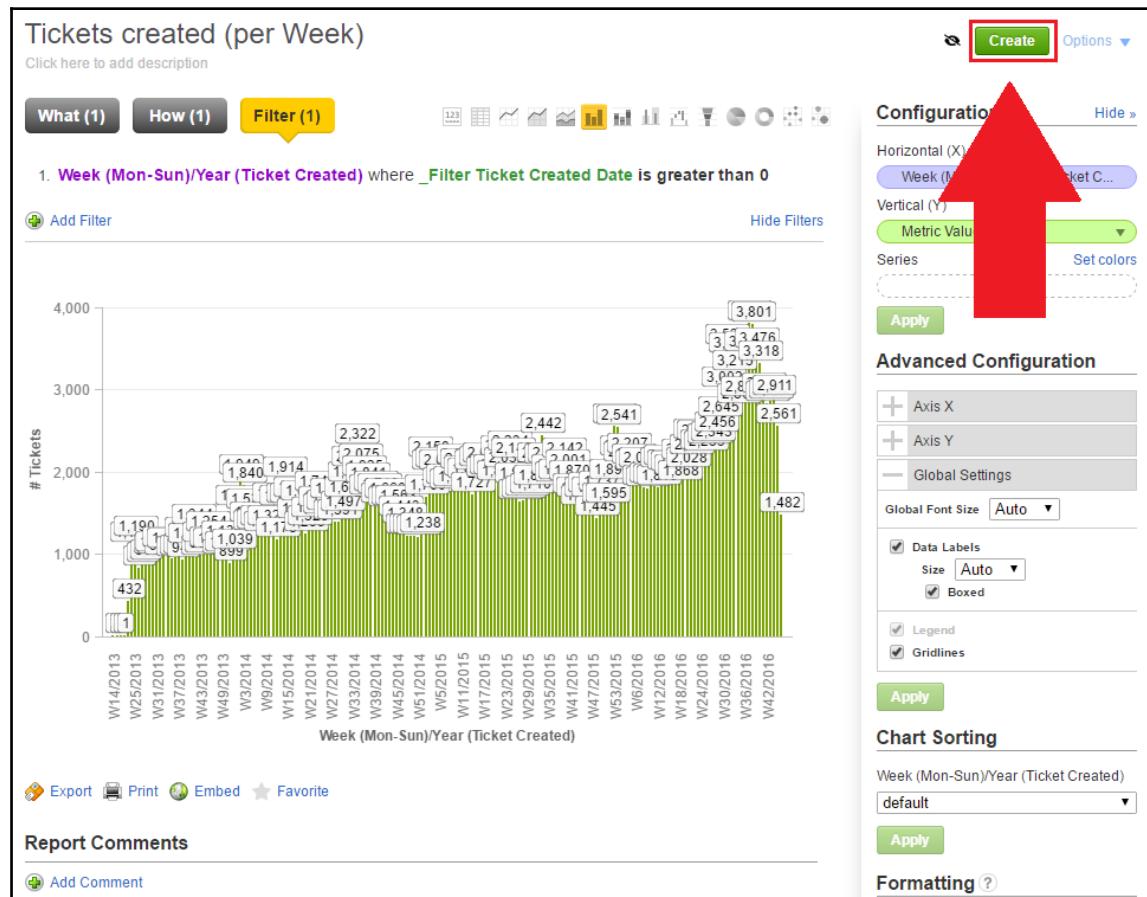
By default, the viewer of the bar chart would have to hover over each bar to review the numeric value. We want to change that and show the value above the bar at all times:

1. Click on **Global Settings** to expand the available settings.
2. Tick the box next to **Data Labels**.
3. Tick the box next to **Boxed**.
4. Confirm by clicking on the **Apply** button:



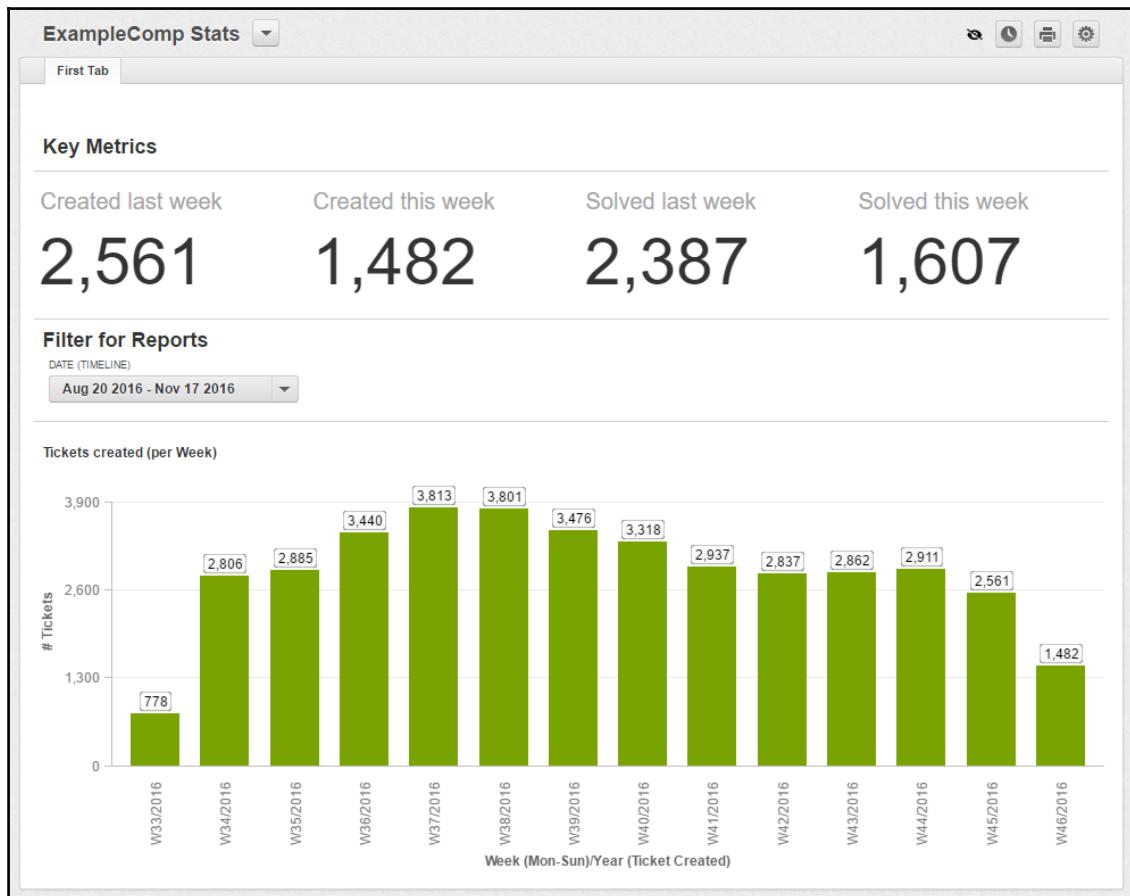
And we are done. Depending on the amount of data, our report might look very cramped. But do not worry, our range filter will take care of that issue when viewing the report on the dashboard.

Create the report by clicking on **Create** in the top right corner:



All we have to do now, is follow the previously learnt steps to add the report to our dashboard's tab.

This is what the result should look like:



Creating reports is a lot of fun and we got plenty of metrics, attributes and filters to play with. Even our ticket-based custom fields will show up as attributes.

I suggest creating another report called **Tickets solved (per Week)** and moving it right below the previously created report for comparison.



Great to know: When editing our dashboard tab, we can edit a selected report by clicking on the little pen icon, which appears above the top-right corner of the report. This also comes in handy, when wanting to review the inner makings of pre-created reports as well.

Creating custom metrics

Creating custom metrics can help to customize your reports even further.

But what does it mean to create a custom metric?

Let us have a look at a possible scenario that would ask for custom metrics as a solution.

Imagine if you want to display the percentage of VIP tickets created within a week.

Without creating your own metric, you could get as far as showing the # of tickets created, with the tag attribute filtered by the VIP tag. You could display that number divided into weeks by using a date filter. Next you could create another report showing the # of all tickets created within the same time frame.

While this solution is definitely not bad, we would have to look at two reports, compare two numbers and do some math to end up with the percentage we were looking for.

So let us start by planning our custom metrics using simple math:

VIP Tickets = Tickets with VIP tag

VIP Tickets / Tickets = Percentage of VIP Tickets as a decimal

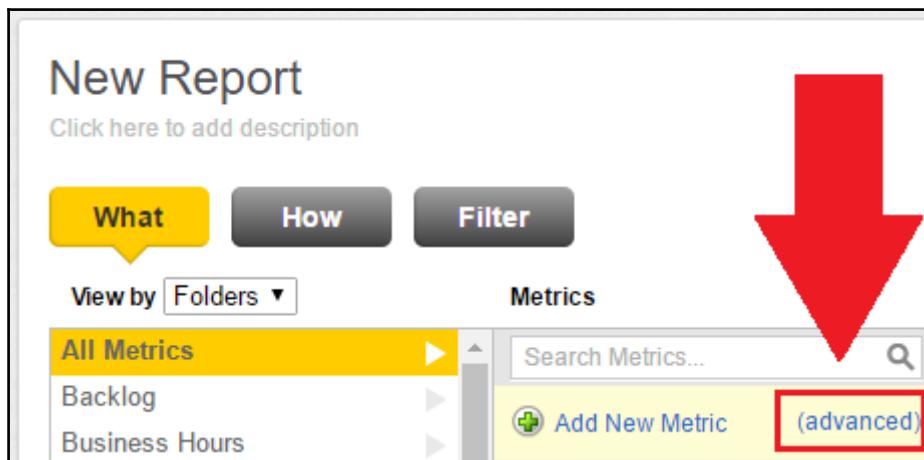
So if we could build a metric called *VIP Tickets* first, we could then use it to create the final metric needed.

Sounds good! But how can we achieve this in GoodData?

GoodData allows us to create new metrics using **Multi-Dimension Analytical Query Language (MAQL)**. A typical MAQL statement would look like this:

```
SELECT # Beans NY Coffee
```

So let us assume, we want to create a new report, but cannot find a suitable metric. GoodData will present us with the option to add a new metric during the creation process of the report itself. Clicking on **Add New Metric** would present us with a more simplified process to create metrics. In our case, we will need to click on **(advanced)**:



Next we can choose between four options:

- **Share (in %)**
- **Difference**
- **Ratio**
- **Custom metric**

The first three options would simplify the creation-process by leading us through the steps of creating a specific type of metric.

Because we want to understand how custom metrics work, we will choose option number four: **Custom metric**.

Share (in%)

... shows a percentage against a fixed whole

Example: month share of Year Revenues

Difference

... shows how a number differs from a fixed part

Example: annual Revenues compared to Year 2005 Revenues

Ratio

... shows a ratio of two previously defined metrics

Example: Revenues to Expenses ratio

Custom metric

Build your own metric using MAQL - a powerful data query language.

Next, GoodData will display the actual MAQL editor for our metric. Before we start writing in MAQL however, let us name our metric and add it to a new folder by the name of Custom Metrics:

The screenshot shows the GoodData MAQL editor interface. On the left, there is a form for defining a metric. The 'Name your metric:' field contains 'CM Vip tickets'. Below it is a checked checkbox for 'Add to Global Metrics'. A dropdown menu 'Folder' is set to 'Create New Folder' with a dropdown arrow, and the folder name 'Custom Metrics' is entered in the adjacent input field. This entire section is highlighted with a red box. To the right of the form is a sidebar titled 'Elements' with a '+' button labeled 'Add Selected'. Under 'Elements', there are several categories: Facts, Metrics, Attributes, Attribute Values, Attribute Labels, and Variables, each with a corresponding grey arrow icon to its right. Below the form, there is a 'Format String:' input field containing '#,##0.00'. Further down, there are tabs for 'Aggregation', 'Numeric', 'Granularity', 'Logical', and 'Filters', with 'Aggregation' currently selected. A detailed description of the 'AVG/RUNAVG - Average, Running Average' function follows, including its definition, examples, and a note about correlation.

Name your metric:
CM Vip tickets

Add to Global Metrics

Folder Create New Folder ▾ Custom Metrics

Format String:
#,##0.00

Aggregation Numeric Granularity Logical Filters

AVG/RUNAVG - Average, Running Average
Returns the average value of all numbers in the set (e.g., Salary paid), Null values are ignored.
Example: `SELECT AVG(Salary Paid)`
Example: `SELECT RUNAVG(Salary Paid)`

CORREL - Correlation
Returns correlation coefficient that varies between -1 and 1.
Correlation indicates the degree of association between two sets of

Back Cancel Add

Finally, we can start by writing our MAQL statement. Let us have a look at our “math statement” again:

VIP Tickets = Tickets with VIP tag

How can we write this statement in MAQL?

First off, whenever using an element such as a metric or an attribute within a statement, we will need to pick it from the element explorer on the right and insert it by clicking on **Add Selected**. Free typing the name of an element will not work. If done correctly, the element will be displayed in color.

And this is what our statement should look like:

```
SELECT IFNULL((SELECT # Tickets WHERE Ticket Tag =  
vip ),0)
```

It will return 1 for each ticket tagged with `vip` and 0 for each ticket without the tag. The following pieces should not be typed, but selected and added via the element explorer:

- `# Tickets` (Metrics)
- `Ticket Tag` (Attributes)
- `vip` (Attribute Values > Ticket Tag)

In order to create the metric, simply click on **Add** located in the lower-right corner:



GoodData will throw us right back to the creation of the report. With one difference. We can now see a Custom Metrics folder with our very own custom metric inside of it:

The screenshot shows the GoodData reporting interface. At the top, there are three tabs: 'What' (highlighted in yellow), 'How', and 'Filter'. Below these, a 'View by' dropdown is set to 'Folders'. The main area is titled 'Metrics' and contains a search bar and a button to 'Add New Metric'. A large red arrow points from the 'Custom Metrics' folder in the left sidebar to the list of metrics on the right. The 'Custom Metrics' folder is highlighted with a red box. The list of metrics includes: Unhappy..., GOLD - First Reply Ti..., GOLD - Full Resolution..., Response Time - Unha..., # Tickets, # Agent Stations, # Assignee Stations, # Backlog Tickets, # Calls abandoned in IVR, # Calls abandoned in on-hold, # Calls abandoned in queue, # Calls abandoned in voice..., and # End Users. At the bottom right are 'Cancel' and 'Done' buttons.

But we are not quite done yet. We did not plan on using this metric in our report. We rather built it to create the actual metric we want to use. So let us repeat the steps that lead us to the MAQL editor the first time and create a second custom metric.

We remember the simple statement:

$VIP\ Tickets / Tickets = \text{Percentage of VIP Tickets as a decimal}$

And turn it into a MAQL statement:

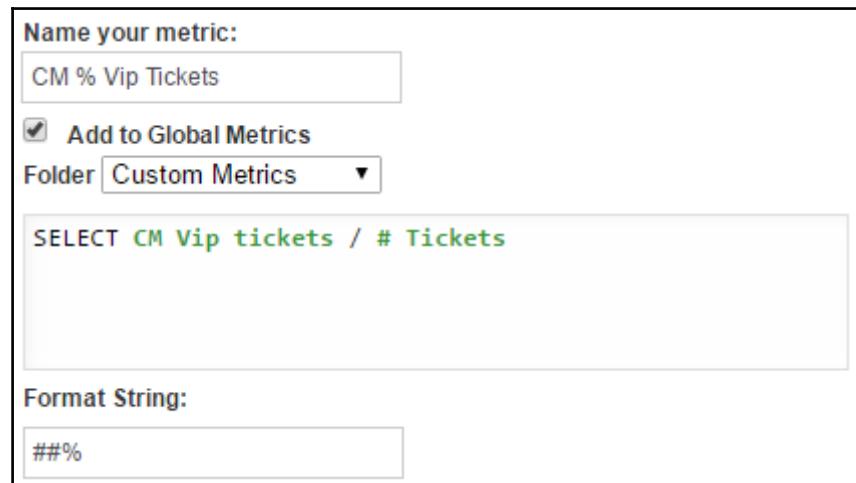
Name your metric:
CM % Vip Tickets

Add to Global Metrics

Folder Custom Metrics ▾

```
SELECT CM Vip tickets / # Tickets
```

Format String:
##%



This time we also changed the **Format String** to ##%. This will assure that the final metric is presented as a two digit percentage.

If we are happy with our metric, we can once again create it by clicking on **Add**.

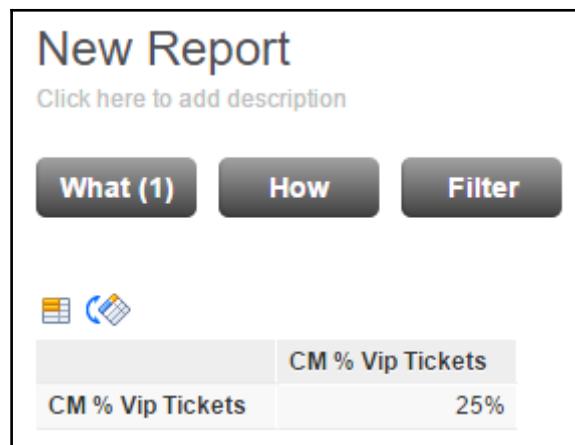
Finally, we only have to choose the new metric as our **What** value. This is what the report should look like:

New Report
Click here to add description

What (1) **How** **Filter**

CM % Vip Tickets

CM % Vip Tickets	CM % Vip Tickets
CM % Vip Tickets	25%



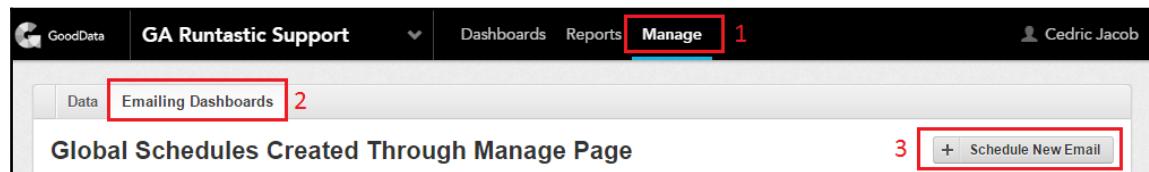
Go ahead and finish the report by adding the **How** and the **Filter** according to what we have learnt before.

Schedule dashboard E-mails

Once we have spent a couple of days or even weeks creating the perfect dashboard for our Zendesk project, we may want to send these reports to other people on a regular basis.

We can do that by scheduling e-mails in GoodData. To schedule an e-mail, please follow the given steps:

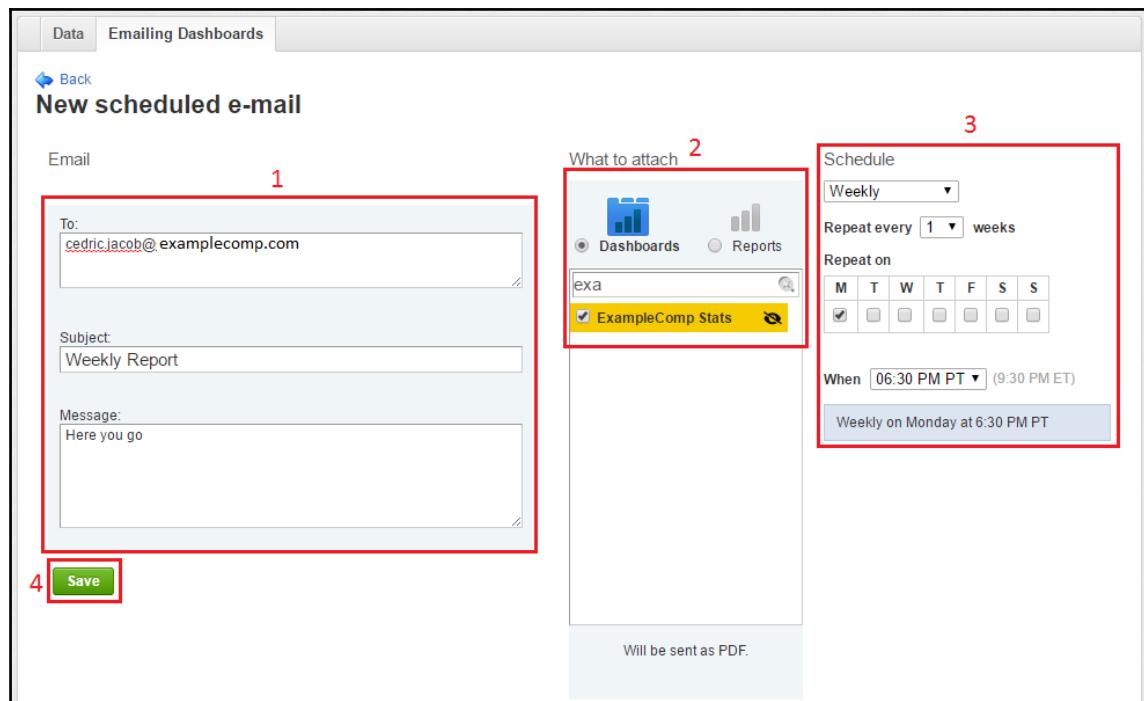
1. Click on **Manage** located in GoodData's top bar.
2. Click on the **Emailing Dashboards** tab.
3. Click on the **Schedule New Email** button.



We will need to supply GoodData with a few pieces of information:

1. Enter recipients' email-addresses, the subject line and a message.
2. Choose a dashboard to send.
3. Pick a schedule for the automated e-mail.

4. Click on **Save**.



Summary

In this chapter we learnt how to utilize Zendesk's very own reporting capabilities. We learnt about GoodData for Zendesk also called Insights and learnt how to create more complex custom reports and metrics.

Creating reports can be a lot of fun and I suggest spending some time on planning your reports. Asking questions is the first step. Using GoodData will give you the answers.

In the next chapter we will go through Zendesk's security settings, cover best practices and learn about SSO.

8

Security Settings and SSO

When communicating with customers, we often deal with highly sensitive and personal information. Naturally, we will have to concern ourselves with the security of our Zendesk setup.

Although Zendesk is ISO 27001:2013 certified, which stands for an established and continuously maintained security system, it is up to us to make use of the many different security settings and features Zendesk has to offer.

In this chapter, we will dive into the world of Zendesk's security, we will review their best practice recommendations, and also take a look at the provided SSO options to authenticate end users as well as agents outside Zendesk.

This chapter covers the following topics:

- Security best practice
- Security settings
- Single sign-on (SSO) options
- Advanced Security Enterprise Add-on

Security settings and best practice

Security is not just a matter of creating a secure system. As long as we, as human beings, have to access and use this system, it becomes our job to uphold security standards as part of our daily routines.

Companies spend millions of dollars on firewalls, encryption and secure access devices, and it's money wasted, because none of these measures address the weakest link in the security chain. – Kevin Mitnick

This does not mean that we should not invest some initial effort in planning and implementing security measurements. It means that security policies, which affect our daily work, should be implemented and upheld as well.

Let's start by looking at the most obvious window of vulnerability—our access security.

Access security

Most likely, agents will log in to Zendesk on a daily basis. It makes sense to review our options and to make this process as secure as possible.

The same goes for our end users. If your Zendesk environment requires end users to log in to create and review their tickets, we should focus on making this process as secure as we can.

Zendesk offers a few different options when it comes to authentication.

By default, your administrators and agents are authenticated and signed in using Zendesk user authentication. You can bypass this and require your administrators and agents to sign in using Google, Microsoft, or a Single sign-on solution using JWT (<https://support.zendesk.com/hc/en-us/articles/203663816>) (Team, Professional, and Enterprise) or SAML (<https://support.zendesk.com/hc/en-us/articles/203663676>) (Professional and Enterprise). – Zendesk

The same goes for end users with the additional option to utilize Twitter and Facebook for social media SSO authentication.

But what does SSO actually mean?

The SSO allows us, the user, to access many systems while actually just logging in to one.

And what does JWT or SAML stand for?

JWT (JSON Web Token) and **SAML (Secure Assertion Markup Language)** are different security token formats used for authentication including SSO.

Zendesk also allows us to add two-factor authorization for agents and admins when using the more simple method of Zendesk authentication. We will cover this option as well.

Before going into more details, let's go ahead and list all the possible options first.

Authentication options for Zendesk agents and administrators:

- Zendesk Password Authentication
- With added two-factor authentication
- SSO Authentication using the following:
 - Google
 - Microsoft
 - JWT
 - SSO

Authentication options for end users:

- Zendesk Password Authentication
- SSO Authentication using the following:
 - Google
 - Microsoft
 - Facebook
 - Twitter
 - JWT
 - SSO

First, let's focus on Zendesk's very own authentication options.

Zendesk password authentication

Zendesk offers three different levels of password security, four if you are an Enterprise customer. In order to adjust this setting, you will need to log in with an admin account.

In order to find and change these settings, follow these steps:

1. Click on the Admin icon (gear symbol) located on Zendesk's sidebar.
2. Click on **Security** located under **Settings** within the admin menu.

3. Click on **Admins & Agents** or **End-users** as per the following screenshot:

The screenshot shows the Zendesk Admin interface. On the left sidebar, under the 'SETTINGS' section, the 'Security' item is highlighted with a red box and labeled '2'. A red box also highlights the 'Admins & Agents' tab in the main content area, which is labeled '3'. The main content area displays the 'Security' settings for Zendesk, stating that administrators and agents sign in using their Zendesk accounts. It offers three password complexity levels: High, Medium, and Low. The 'High' level is selected.

Level	Requirements
High	<ul style="list-style-type: none">must be different than the previous 5 passwordsmust be at least 6 charactersmust be different from email addressmust include letters in mixed case and numbersmust include a character that is not a letter or number10 attempts allowed before lockoutexpires after 90 days
Medium	<ul style="list-style-type: none">must be at least 6 charactersmust be different from email addressmust include letters in mixed case and numbersmust include a character that is not a letter or number10 attempts allowed before lockout
Low	<ul style="list-style-type: none">must be at least 5 charactersmust be different from email address10 attempts allowed before lockout

Let's take a closer look at the three standard options:

- **High**

- must be different from the previous 5 passwords
- must be at least 6 characters long
- must be different from e-mail address

- must include letters in mixed case and numbers
 - must include a character that is not a letter or number
 - 10 attempts allowed before lockout
 - expires after 90 days
- **Medium**
 - must be at least 6 characters long
 - must be different from e-mail address
 - must include letters in mixed case and numbers
 - must include a character that is not a letter or number
 - 10 attempts allowed before lockout
 - **Low**
 - must be at least 5 characters long
 - must be different from e-mail address
 - 10 attempts allowed before lockout

When using Zendesk Authentication, we should definitely stay away from option 3 (low) and consider the highest security level instead. Having each password expire after 90 days adds another layer of security.

In addition to choosing the highest security level, agents should be made aware of the following:

- One should not use the same password for different services.
- Random generated strings are almost impossible to guess and therefore the best option.
- Consider using a keychain or a password manager. Having to enter the password every single time makes us more vulnerable to key loggers.
- When choosing a new password, do not simply add a number at the end.

If you are on the Enterprise plan, consider using option number 4 **Custom**, which allows you to pick your own requirements:

The screenshot shows a configuration interface for custom password requirements. It includes a title 'Custom' with a radio button icon, a list of requirements with dropdown menus for selection, and a 'none' option for the first requirement.

Requirement	Setting
must be different than at least this many previous passwords:	none
must be at least this many characters:	6
must include numbers and special characters:	numbers and special characters
must include letters in mixed case:	yes
expires after how many days:	never
number of failed attempts until lockout:	10
sessions expire after how many minutes:	8 hours
max number of consecutive letters or numbers allowed:	unlimited
must not include the local part of the user's email (localpart@domain.com):	no

Here are my preferred **Custom** settings:

- Must be different from at least these many previous passwords: **7 previous passwords**
- Must be of at least these many characters: **12**
- Must include numbers and special characters: **numbers and special characters**
- Must include letters in mixed case: **yes**
- Expires after how many days: **90**
- Number of failed attempts until lockout: **3**

- Sessions expire after how many minutes: **8 hours**
- Max number of consecutive letters or numbers allowed: **3**
- Must not include the local part of the user's e-mail (`localpart@domain.com`):
yes

Finally, we may choose to enable Zendesk's own two-factor authentication, which only works when using Zendesk Authentication.



Something to consider: When using SSO, you may still choose to use the two-factor authentication utilizing your own or third-party services.

Two-factor authentication

Two-factor authentication adds another level of protection by asking agents and admins to enter an expiring code when logging in to Zendesk.

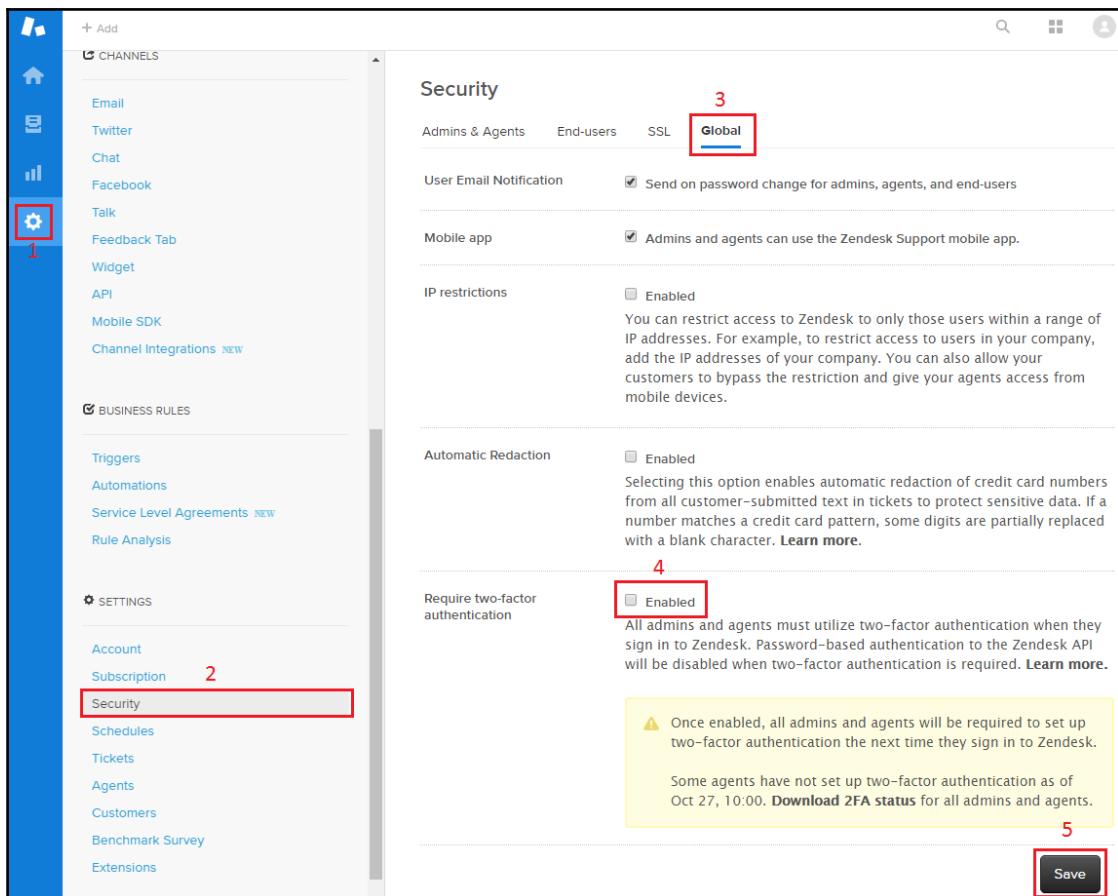


Note that while this works great for logging in via the website or Zendesk's mobile apps, you will run into an issue when trying to use the Zendesk REST API, which does not currently support the two-factor authentication. In that case, you will need to use an API token or make use of the OAuth authentication flow. Both are explained in detail in Zendesk's developer documentation.

In order to enable the two-factor authentication, follow these steps:

1. Click on the Admin icon (gear symbol) located on Zendesk's sidebar.
2. Click on **Security** located under **Settings** within the admin menu.
3. Navigate to the Global Settings tab, by clicking on **Global**.
4. Tick the box next to Require two-factor authentication.

5. Click on **Save** to submit the change:



So how does this work for our agents?

Next time an agent logs in, they will have to set up their two-factor authentication as required. They can choose to either use a mobile app (such as Google Authenticator) or the more conventional SMS method.

The process is straightforward and easy to understand.

Next, let's review Zendesk's Single sign-on options.

Zendesk's single sign-on options

On top of using the authentication process offered by Zendesk, we have the option to use SSO as well.

You already learned a little bit about SSO or Single sign-on, which allows us to authenticate users in one system, leading to automatically authenticating them in other systems as well.

What does that mean for us?

By adding SSO to our environment we can authenticate users outside Zendesk.

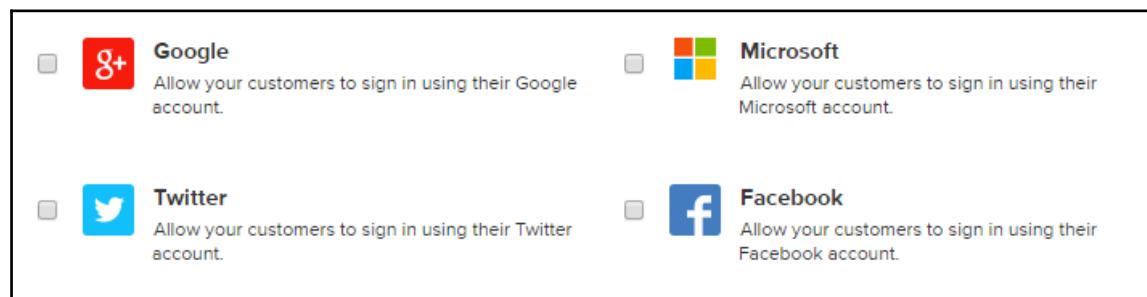
Zendesk offers two different types of SSO:

- Social media SSO
- Enterprise SSO

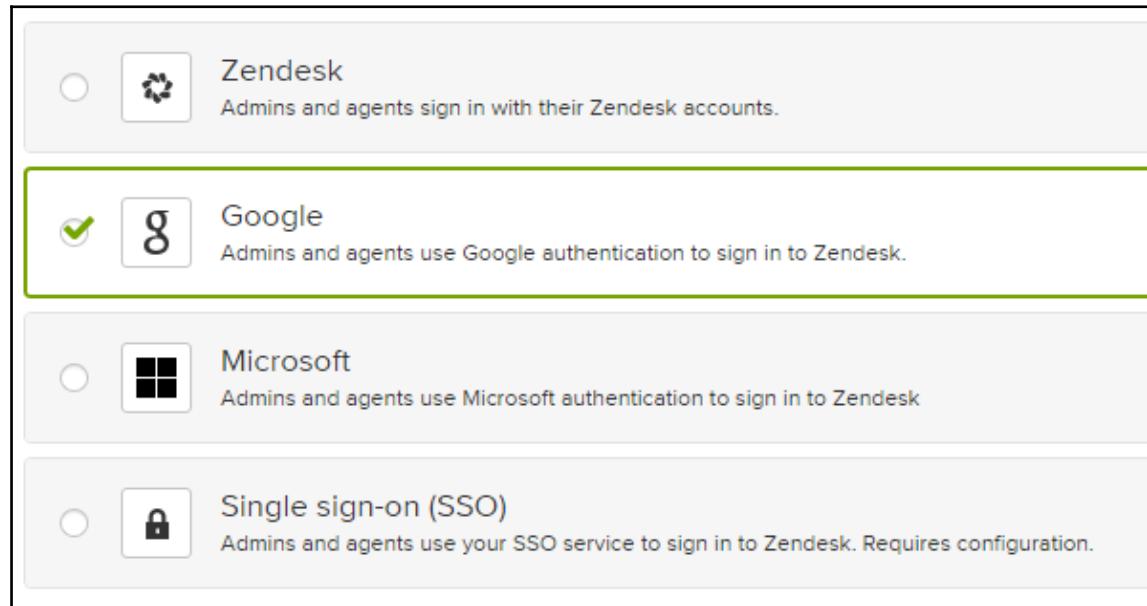
Social media SSO

Social media SSO is great for end-users and functions as an additional option for customer convenience. Zendesk enables us to add Facebook, Google, Microsoft, and Twitter as login options for end users, all of which will work on top of the standard Zendesk Authentication.

Just below the security level settings, we find the following:



For agents, we can choose either Google or Microsoft, with one distinct difference; it will replace the Zendesk's password authentication altogether. So instead of being able to add these options, choosing either one of them will deactivate the others:



So far so good. In some cases, we might want to use another type of SSO and not make use of existing social media options. This is where Enterprise SSO comes into play.

Enterprise SSO

While it is possible to enable Enterprise SSO for only agents or only end users, Enterprise SSO will replace any other login options for the respective type of user.

When using Enterprise SSO, the authentication happens outside Zendesk.

As mentioned before, there are two different options when it comes to Enterprise SSO:

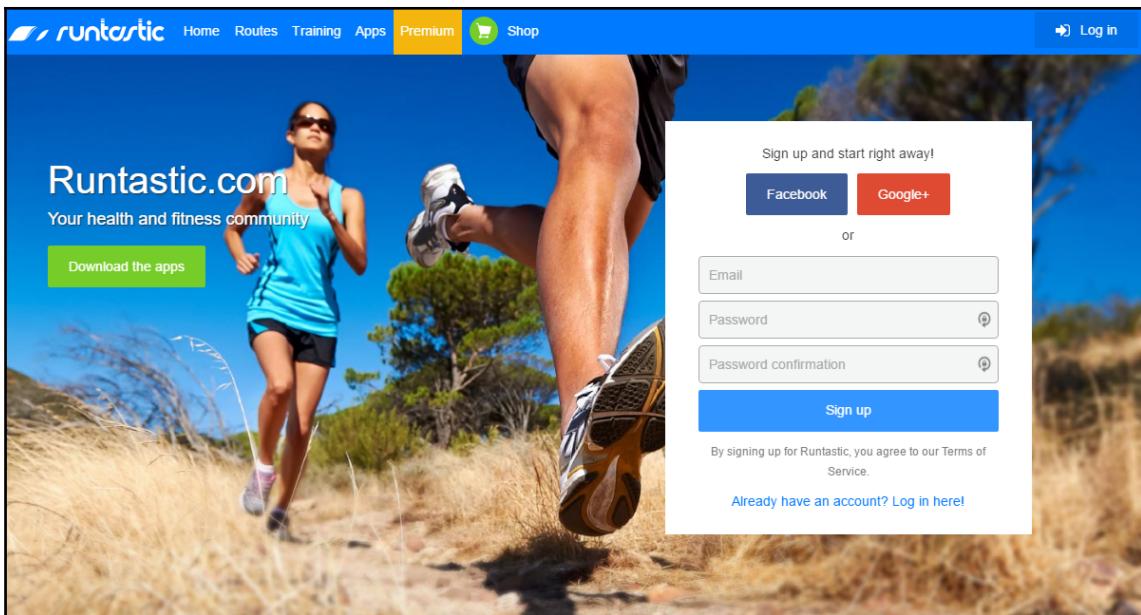
- **JSON Web Token (JWT)**
- **Secure Assertion Markup Language (SAML)**

We will discuss both these options in more detail later on.

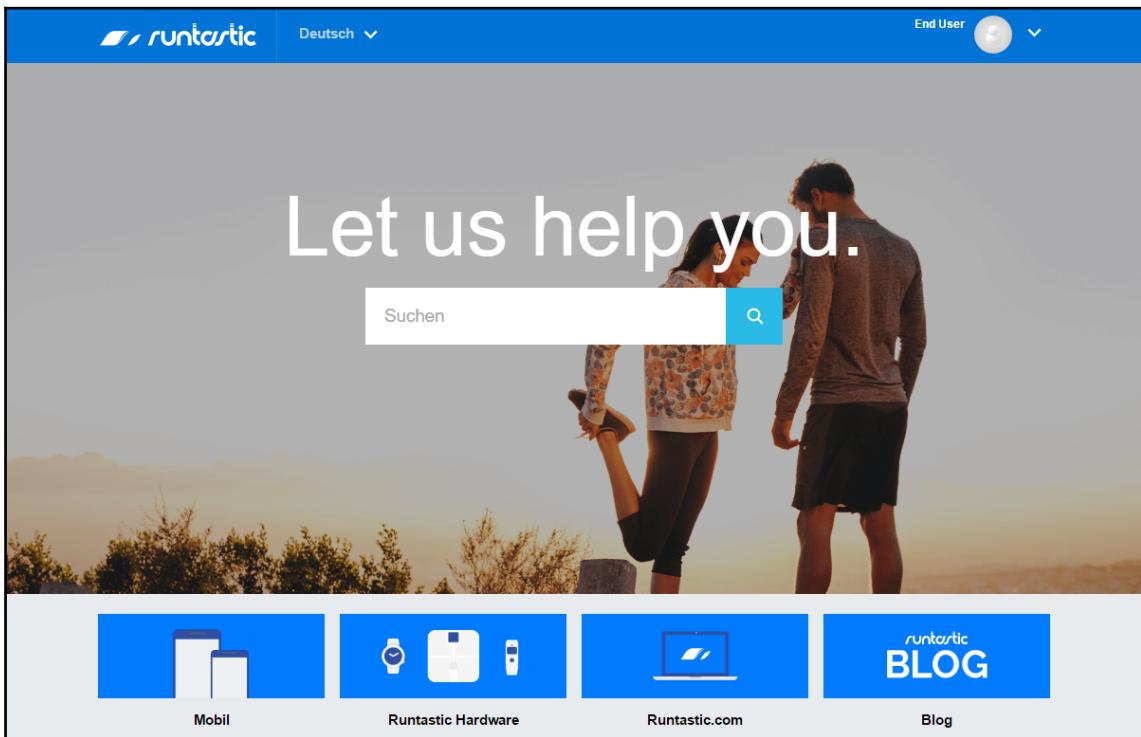
Great, but how does this work for our agents and/or end-users?

Good question! Let's look at a real-life example to shed some light on the practical side of things.

Runtastic, an Austrian company, is using the Enterprise SSO option:



While it looks like any other login/sign-up page, if a user, once signed in, decides to click on their support-link, they will be forwarded to Runtastic's Help Center:



When reviewing the preceding screenshot, you will notice that the user is actually logged in and therefore authenticated. The user did not have to sign up or log in separately.

Due to their SSO solution, upon logging in to their system, Zendesk was subsequently told that the user has been authenticated.

That is it. It may not seem like much, but that is the point entirely. Making it more comfortable for the customer while upholding a higher security standard, namely our own.

What would the other options be?

Well, we could allow end users to contact us via the Help Center without being logged in. The end user will have to provide their e-mail address via the contact form. This is fine for simple questions, such as how do I use your service, but as soon as the account changes or sensitive information come into play, we will have to verify that we are actually conversing with the user in question.

We can also force the user to sign-up upon visiting our Help Center. While this will fix our security concerns as the user will have to verify their e-mail during the sign-up process, it would not be as comfortable as we would like it to be.

JSON Web Token (JWT)

Implementing JWT is surely a job for a seasoned developer. Zendesk offers a range of examples when it comes to the implementation of JWT, which are accessible through their documentation online.

Understanding the process can be of advantage for Zendesk admins as well as developers who have not yet dealt with JWT and/or Zendesk before. So let's take a closer look at the process:

1. A user navigates to our Help Center (<https://support.examplecomp.com>), which realizes that the user is not yet authenticated.
2. The Help Center redirects the user to our specified URL (<https://www.examplecomp.com/login>) for authentication.
3. The user is authenticated using our login process and a JWT request is built using user-related data from our system.
4. Now, the user is redirected to Zendesk (<https://examplecomp.zendesk.com/access/jwt>) with the JWT payload.
5. Zendesk checks the JWT payload and authenticates the user.

This is the beauty of the whole process. There is no direct connection between Zendesk and our system. Our system simply hands over some data, and through some redirects, the user ends up authenticated as well as at the location intended.

Sounds good, but what is a payload?

The term payload, simplified, could be translated to the word “message”.

So what does our payload look like?

The payload, which must be appended to the target URL (the endpoint: <https://examplecomp.zendesk.com/access/jwt?jwt=...>) as a query string, should be a base64-encoded hash and carry at least four compulsory attributes.

A query string can be used to hand over information to a website. Here is an example:

<https://www.PleaseTellMeYourNameExample.com/?name=Cedric>

This is obviously not a real website, but it should help to clarify the term.

So the mentioned payload is a message, containing user data, which is encoded and appended to the target URL for Zendesk to review the data and authenticate the user.

How do we build the payload?

Well, the payload consists of a header and the payload's actual content. Let's review each part and how it is put together.

The header

Here is what the header will look like:

```
{  
  "typ": "JWT",  
  "alg": "HS256"  
}
```

The structure might look familiar to you. This is because we dealt with the JSON structure previously in Chapter 6, *Integrating and Extending Zendesk*.

The first key and its value make sense. In our header, we declare the type of this request, which is `JWT`.

The second key is the algorithm we want to use. The corresponding value `HS256` stands for HMAC SHA 256, an encryption algorithm developed by the NSA.

The content

Let's focus on the four mandatory attributes first. Only using these, the content would look something like this:

```
{  
  "iat": 1477621409,  
  "jti": 3843566535171.367,  
  "email": "genericperson@example.org",  
  "name": "Generic Person"  
}
```

The third and fourth key make sense. Zendesk uses the e-mail address as a unique identifier to find the user in the system. The `name` key is necessary to create the user or to update it later on.

But what are the first two keys?

The `jti` key stands for **JSON Web Token ID**, which consists of a unique ID, basically consisting of a random string. It is meant to prevent token replay attacks by making each request unique. If a hacker were to watch your traffic, they will not be able to simply use the same URL to access the account.

The `iat` key stands for **Issued At** and consists of a timestamp, formatted using the UNIX standard that describes the exact time when the JSON Web Token was issued. This timestamp ensures that the token is only used shortly after its creation.

While these four keys are mandatory, we could decide to add a number of other keys corresponding with the user fields in Zendesk. We remember that when contemplating how to add the VIP tag for a user, we had the following idea:

We would set the user tag and subsequently the ticket tag via SSO in Chapter 2, Agent Roles, Groups, Organizations, and User Tags.

So, in our script that builds the payload, we should add the necessary condition. If the user is in fact a VIP customer of ExampleComp, we can set the user's tags and consequently mark them as a VIP user in Zendesk.

So, if a VIP customer is logging in, this is what our content will look like:

```
{  
    "iat":1477621409,  
    "jti":3843566535171.367,  
    "email":"genericperson@example.org",  
    "name":"Generic Person",  
    "tags":"vip"  
}
```

Great. Now that we have got the header and the content sorted out, we can go ahead and create our payload, right?

Well, almost. However, before we can do that, we need to know a secret.

The Secret/enabling SSO in Zendesk

The secret is a generated token, which we receive upon enabling JWT SSO in Zendesk. So let's go ahead and do that:

1. Click on the Admin icon (gear symbol) located on Zendesk's sidebar.
2. Click on **Security** located under **Settings** within the admin menu.
3. Navigate to the **Admins & Agents** or **End-users** tab (you can set up JWT SSO for both or just one of them).
4. Click on **Single sign-on (SSO)** and check the box next to **JSON Web Token**.
5. Enter **Remote login URL** and **Remote logout URL**.
6. Click on **Generate a new token**. This is our secret, make sure to copy it. Zendesk will hide it later on.
7. Click on **Save**:

The screenshot shows the 'JSON Web Token' configuration page in Zendesk. The 'JSON Web Token' checkbox is checked, and a descriptive text about JWT is displayed. The 'Remote login URL' field contains 'https://'. Below it, a note says 'This is the URL that Zendesk will redirect your users to for remote authentication, e.g. https://www.example.com/services/login'. The 'Remote logout URL' field contains 'https://'. Below it, a note says 'This is the URL that Zendesk will redirect your users to after they sign out, e.g. https://www.example.com/services/logout'. The 'IP ranges' field is empty. Below it, a note says 'Requests from these IP ranges will always be routed via remote authentication. Requests from IP addresses outside these ranges will be routed to the normal sign-in form. To route all requests through remote authentication, leave this blank. An IP range is in the format n.n.n.n, where n is a number or an asterisk (*) wild card. Separate multiple IP ranges with a space. Your current IP address is: 91.59.125.254'. The 'Update of external IDs?' dropdown is set to 'On'. Below it, a note says 'It is safe to ignore this setting if you do not use external_id's. When enabled, the external_id of the user being signed in can be updated. This only happens when a user with the external_id is not found, but the user's email address is found. The external_id is unique for an account. Users without an external_id will have one added if it's present in the authentication request.' The 'Shared secret' section shows the first 6 characters of the token: 'SHU1Gc'. A note says 'For security reasons, we only display the first 6 characters of your existing remote authentication token here:'. Another note says 'If you need to reset the token, click this link: [Regenerate](#)'. A final note at the bottom says 'If you generate a new token, be sure to update your local authentication script. Note that the token is not updated until you submit this form.' At the very bottom, a note says 'The token is a shared secret between you and Zendesk. It must never be publicized.'



Very important: Keep the secret safe! Exposing the shared secret will compromise your account.

Now that we know the secret, we can start putting it all together.

Putting it together

Let's take a look at one of Zendesk's example implementations written in Ruby:

```
require 'securerandom' unless defined?(SecureRandom)

class ZendeskSessionController < ApplicationController
  # Configuration
  ZENDESK_SHARED_SECRET = ENV["ZENDESK_SHARED_SECRET"]
  ZENDESK_SUBDOMAIN    = ENV["ZENDESK_SUBDOMAIN"]

  def create
    if user = User.authenticate(params[:login], params[:password])
      sign_into_zendesk(user)
    else
      render :new, :notice => "Invalid credentials"
    end
  end

  private

  def sign_into_zendesk(user)
    iat = Time.now.to_i
    jti = "#{iat}/#{SecureRandom.hex(18)}"

    payload = JWT.encode({
      :iat    => iat,
      :jti    => jti,
      :name   => user.name,
      :email  => user.email,
    }, ZENDESK_SHARED_SECRET)

    redirect_to zendesk_sso_url(payload)
  end

  def zendesk_sso_url(payload)
    url = "https://#{ZENDESK_SUBDOMAIN}.zendesk.com/access/jwt?jwt=#{payload}"
    url += "&return_to=#{URI.escape(params["return_to"])}" if
    params["return_to"].present?
    url
  end
end
```

Even with limited knowledge in programming, we can try to understand what is going on here. Taking the code apart might make it a little easier for us:

```
ZENDESK_SHARED_SECRET = ENV["ZENDESK_SHARED_SECRET"]
ZENDESK_SUBDOMAIN = ENV["ZENDESK_SUBDOMAIN"]
```

This part takes care of the configuration. It allows us to provide our shared secret and our subdomain:

```
def create
  if user = User.authenticate(params[:login], params[:password])
    # If the submitted credentials pass, then log user into Zendesk
    sign_into_zendesk(user)
  else
    render :new, :notice => "Invalid credentials"
  end
end
```

This function authenticates the user in our system and checks whether the credentials are all right. If they are, it calls the `sign_into_zendesk(user)` function, passing on the user data:

```
def sign_into_zendesk(user)
  iat = Time.now.to_i
  jti = "#{iat}/#{SecureRandom.hex(18)}"

  payload = JWT.encode({
    :iat => iat,
    :jti => jti,
    :name => user.name,
    :email => user.email,
  }, ZENDESK_SHARED_SECRET)

  redirect_to zendesk_sso_url(payload)
end
```

This function generates the `iat` and `jti` values first, before creating payload by encoding the four necessary attributes utilizing the shared secret. Once the payload is ready, it moves to `redirect_to` and calls the following function to generate the URL according to the `zendesk_sso_url(payload)` payload:

```
def zendesk_sso_url(payload)
  url =
"https://#{ZENDESK_SUBDOMAIN}.zendesk.com/access/jwt?jwt=#{payload}"
  url += "&return_to=#{URI.escape(params["return_to"])}" if
params["return_to"].present?
  url
```

end

This function generates the URL by piecing the following together:

- Zendesk's subdomain
- The payload
- The return address

Makes sense, but where does the return address come from?

As soon as Zendesk redirects a user to the remote authentication site, it automatically adds a `return_to` parameter to the URL. All we have to do is grab that return address and add it to our URL as well.

You may want to check out some of the other JWT examples on GitHub at https://github.com/zendesk/zendesk_jwt_sso_examples.

JWT is great, but what about SAML?

Secure Assertion Markup Language (SAML)

SAML for Zendesk works somehow similar to the JWT approach. Before having a closer look at the implementation, let's try to understand how SAML works first.

Security Assertion Markup Language, also called SAML, is a data format used for authenticating data between different parties. It is based on XML and mainly used to authenticate for login.

The SAML authentication process would look like:

- The user tries to log in to a Zendesk service like our Help Center
- The user is redirected to our SAML server for authentication
- The SAML server sends a request to our Identity Directory
- Once the user is authenticated, the user is redirected to the Zendesk service:



So, what we need is an **SAML** server and **Identity Directory**. It is also possible to store the user data straight on the SAML server, which will render Identity Directory unnecessary. This heavily depends on your current setup.

But what is an Identity Directory?

The term “Identity Directory” refers to the system that holds our user's data. There is not one Identity Directory, but many different solutions and options. One example is the Active Directory by Microsoft.

To make it a little less complicated, let's assume that ExampleComp wants to store all their user data on their SAML server and we do not need an additional directory.

Great, and how do we get an SAML server?

We can choose to set up our own SAML server utilizing OpenAM for example. Alternatively, we could choose to go with one of the many existing services such as OneLogin or PingIdentity.

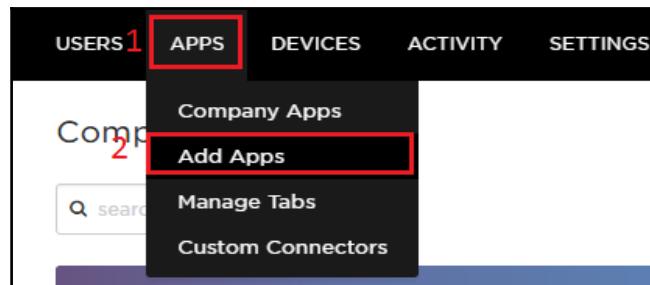
Let's go through the steps of implementation before tackling them one by one:

1. First, we will set up our SAML sever and take note of the remote login URL. We will also need either the SHA1 or SHA2 fingerprint of our server's SAML certificate.
2. Next, we will use the login URL and the fingerprint to enable SAML in our Zendesk setup.

Setting up our SAML server

Luckily for us, OneLogin provides a simple solution when it comes to connecting Zendesk. Simply follow the steps outlined here:

1. Log into your OneLogin account (or create one if necessary) and click on **APPS** located within the top bar:



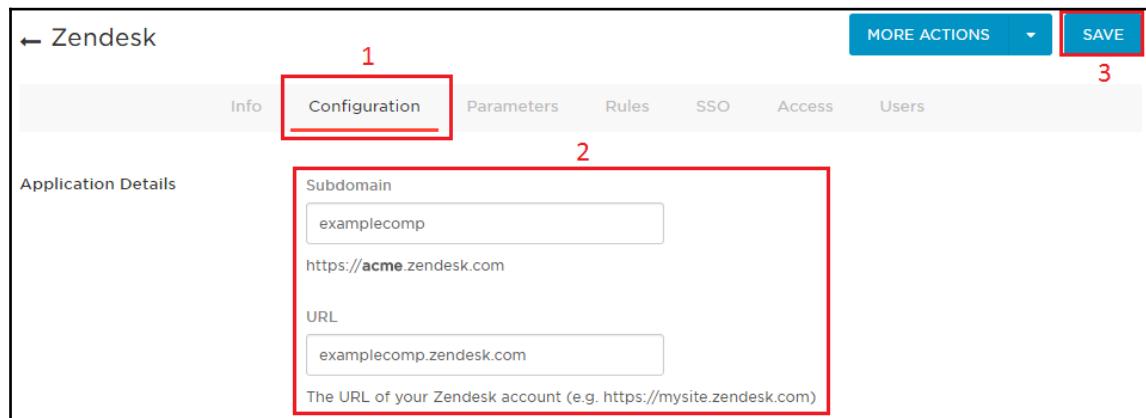
2. Enter **Zendesk** in the search field and click on the search result that lists SAML in the description:

The screenshot shows a search interface titled 'Find Applications'. A red box labeled '1' highlights the search input field containing 'Zendesk'. A red box labeled '2' highlights the search result for 'Zendesk' from 'Zendesk, Inc.', which includes a small icon, the name, and a brief description: 'Form-based auth, OpenID, SAML2.0 , browser extension , provisioning'.

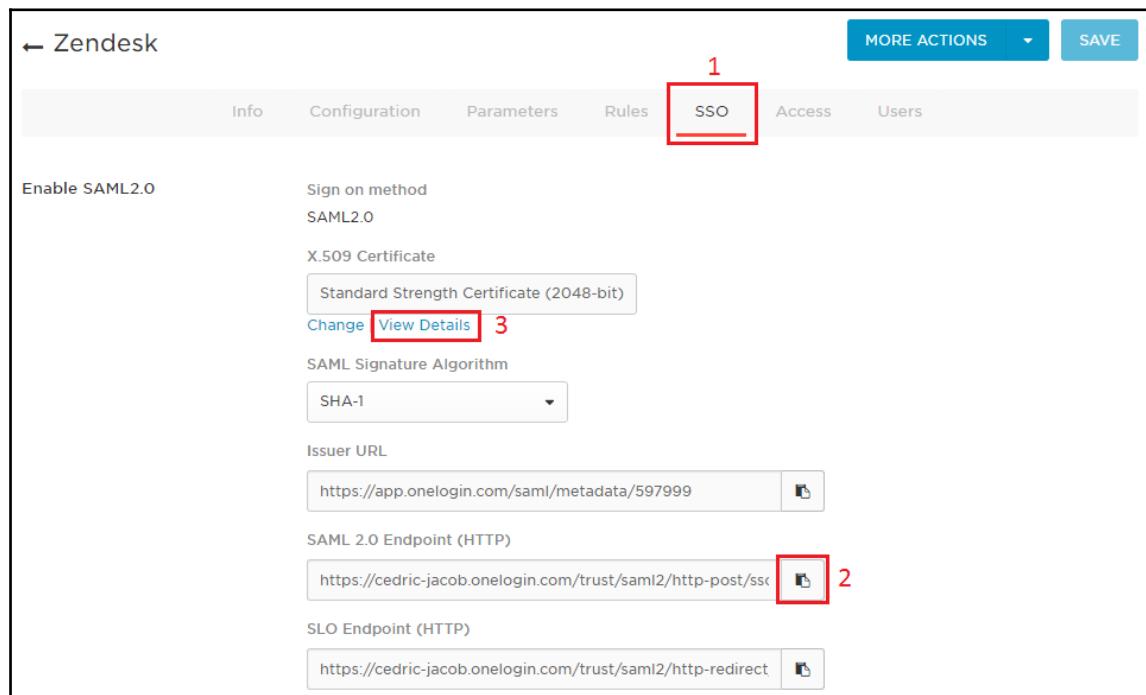
3. Choose SAML 2.0-user provisioning and click on **SAVE**:

The screenshot shows the 'Add Zendesk' configuration page. A red box labeled '1' highlights the 'Connector Version' section under 'Connectors', where the 'SAML2.0 - user provisioning' option is selected. A red box labeled '2' highlights the blue 'SAVE' button in the top right corner. The page also includes sections for 'Portal' (Display Name: Zendesk, Visible in portal: checked), 'Rectangular Icon' (an uploaded logo image), and 'Square Icon' (a placeholder for a square icon).

4. Click on the **Configuration** tab, enter your Zendesk's **Subdomain** and full URL, and click on **SAVE**:



5. Switch to the **SSO** tab, take note of the **SAML 2.0 Endpoint (HTTP)**, and click on **View Details**:



6. Take note of **SHA fingerprint**:

A screenshot of the Zendesk Admin interface under 'Security Settings'. It shows a certificate named 'Standard Strength Certificate (2048-bit)' with a key length of '2048-bit'. The 'SHA fingerprint' section is highlighted with a red box around the 'Fingerprint' input field, which contains the value '03:B5:0B:FA:C6:D'. There are 'DELETE' and 'SAVE' buttons at the top right.

Having taken note of all the necessary information, we can get back to Zendesk.

Enabling SAML in Zendesk

Finally, we will have to enable SAML in Zendesk:

1. Click on the Admin icon (gear symbol) located on Zendesk's sidebar.
2. Click on **Security**, located under **Settings** within the admin menu.
3. Navigate to the **Admins & Agents** or **End-users** tab (you can set up SAML SSO for both or just one of them).
4. Click on **Single sign-on (SSO)** and check the box next to **JSON Web Token**.
5. Enter **SAML SSO URL** and **Certificate fingerprint**.
6. Click on **Save**:

A screenshot of the Zendesk Admin interface under 'Single sign-on (SSO)'. The 'SAML' section is selected. It includes fields for 'SAML SSO URL' (set to 'https://'), 'Certificate fingerprint' (empty), 'Remote logout URL' (empty), and 'IP ranges' (empty). A note at the bottom explains IP range syntax. A green checkmark icon is present on the left.

While enabling SSO can surely add a great layer of security to our setup, we may want to look into IP restrictions to make it even harder to penetrate our system.

IP restrictions

Restricting access to Zendesk to specific IPs is also an option. This can be done for agents and end users alike.

To enable this option, navigate to **Security** settings and switch to the **Global** tab.

Zendesk gives us the option to specify the IPs that should receive access. This can be done by entering specific IPs and/or entering IP ranges. We can also let end users bypass these restrictions and allow agents who use the mobile Zendesk app to bypass it as well:

IP restrictions Enabled

You can restrict access to Zendesk to only those users within a range of IP addresses. For example, to restrict access to users in your company, add the IP addresses of your company. You can also allow your customers to bypass the restriction and give your agents access from mobile devices.

Allowed IP ranges

Separate multiple IPs with a space

Only requests from these IP ranges will be allowed. Specify the IP range using asterisk (*) wild cards in n.n.n.n (where n is a number or an asterisk) or using the **IP subnet mask** syntax n.n.n.n/n. Separate multiple IP ranges with a space. [Learn more](#). Your current IP address is: 91.59.127.128

⚠️ Enabling IP based access restrictions can break third-party integrations. Be sure to whitelist all external IPs that need access to your account via the Zendesk API.

Customers can bypass restrictions
Select this option to allow your customers to access Zendesk outside the IP ranges.

Agents can bypass IP restrictions using mobile apps
Select this option to allow agents to access Zendesk Support from native Zendesk Support apps (iPhone, iPad & Android) regardless of IP address.



Note that in one of the previous chapters, we covered Zendesk integrations and extensions. If some of the third-party services need access to Zendesk, make sure to whitelist their external IPs as well!

Great access security can only do so much. If someone were to gain access by hijacking one of our accounts, we may want to reduce their scope of action.

Using custom roles to reduce unnecessary access

One great way to ramp up your security is utilizing custom roles in Zendesk.

How?

Give agents only those permissions that they really need.

The logic behind this idea is very simple. If a hacker were to gain access to one of your agent accounts, their scope of action will be limited.

Even more important is to have as little accounts with admin permissions as possible. If someone happens to gain access to an admin account, they will have the necessary permissions to change security settings as well.

Reviewing the permissions of your accounts and thereby reducing as much risk as possible goes a long way.

Sometimes we may not even notice an intruder at all. A great way to make sure that our system is not comprised is to review it constantly.

Auditing your Zendesk account

Besides encouraging your agents to review their account activity, auditing your Zendesk account should become regular practice. Here are a few pointers:

- **Review your agents and admin accounts:** Keep an eye out for any changes that may have been made to the accounts. Check for unknown e-mail addresses that may have been added. Review the assigned roles and suspend access of agents who are on long holidays or recently stopped working.

- **Review your business rules:** Make sure that no new business rules have been added. Check whether your notifications are going out to the right users.
- **Review Zendesk's security settings:** It is a good idea to go through your security settings such as IP restrictions as well. Any changes could point to possible breaches.
- **Make use of your account audit logs:** If in doubt, the Enterprise version of Zendesk offers the option to monitor security events via the Audit log feature. Any changes are logged and can be reviewed later at any given time.

The auditing should not stop there. What about the security of our apps?

Secure coding practices

At the risk of repeating myself: the topic of secure coding practices could definitely fill another book.

However, there are a few pointers when it comes to extending Zendesk using the REST API:

1. If you do not use the REST API, disable it on the Channels page.
2. If you do use the Zendesk API, using OAuth authentication (via token) is more secure, and therefore the preferred option.
3. Make sure to keep your apps updated and remove them as soon as they become redundant.
4. General secure coding practices apply.

No system will ever be totally safe. If someone were to access our account, we probably would not want them to find sensitive data.

Removing/Redacting Credit Card numbers

When dealing with customers, from time to time end-users might decide to include credit card information in their communication. This information is then stored with the ticket even though it should not be accessible to us later on.

Zendesk allows us to enable Automatic Redaction in order to take care of this issue.

To enable this option, navigate to **Security settings**, switch to the **Global** tab, and tick the box next to **Automatic Redaction**:

Automatic Redaction	<input type="checkbox"/> Enabled Selecting this option enables automatic redaction of credit card numbers from all customer-submitted text in tickets to protect sensitive data. If a number matches a credit card pattern, some digits are partially replaced with a blank character. Learn more.
----------------------------	--

Private attachments

When end-users send attachments, they are saved on a server and therefore accessible to everybody who knows the exact link. While the attachments are not indexed by search engines, the fact that they are indeed public should be a valid concern to us.

Zendesk gives us two options to deal with this problem:

1. We disable attachments all together.
2. We can enable private attachments and require users to authenticate before downloading an attachment.

While disabling attachments might be safer, allowing end-users to provide images or other files may be essential to offer effective support.

In order to enable private attachments or disable attachments, follow these steps:

1. Click on the Admin icon (gear symbol) located on Zendesk's sidebar.
2. Click on **Tickets**, located under **Settings** within the admin menu.
3. Navigate to the Settings tab, by clicking on **Settings**.

4. Scroll down to the **Attachments** settings:

The screenshot shows the Zendesk Tickets settings page. On the left sidebar, under the 'SETTINGS' section, 'Tickets' is highlighted with a red box and labeled '2'. In the main content area, the 'Settings' tab is selected and highlighted with a red box and labeled '3'. The 'Attachments' section is also highlighted with a red box and labeled '4'. The 'Attachments' section contains the following configuration options:

- Customers can attach files: Allows end-users to submit attached files. Covers attachments by email too, and does not apply to agents.
- Require authentication to download: Enables higher security for attachments by requiring users to sign in to view ticket attachments.

Advanced Security Enterprise Add-on

At additional cost (per agent), Zendesk offers an Advanced Security Add-on for their Enterprise customers. Let's take a quick look at the advantages that come with the purchase.

Data at rest encryption

In order to provide additional protection for our stored data, Zendesk will encrypt inactive data, which includes the following:

- User, ticket, and Help Center data
- Search data
- Logs
- Backups
- Attachments

Enhanced disaster recovery

Zendesk will perform daily backups of all our data. In case of a disaster, we can recover the lost data. The option to switch to another data center (in the same region) will also be available.

HIPAA compliance

Zendesk assures compliance with the Health Insurance Portability and Accountability Act.

You can learn more about the HIPAA by visiting the following website: <http://www.hhs.gov/hipaa/index.html>

In case of a breach

Taking security seriously is important. However, even after following every best practice guide and applying yourself every day, our system may be breached.

What now?

As soon as we start suspecting a breach, we should get in contact with Zendesk. There are a few ways to do this:

- We can create a ticket using the word “Security” in the subject line
- We can send an e-mail to security@zendesk.com
- We can call Zendesk's customer support:
 - Americas/US: 415-418-7506
 - Europe/UK: +44 20 3355 7960
 - Asia-Pacific/Australia: +61 3 9008 6775

In each case, we should make sure to provide all the necessary details including our thought process that lead us to believe a breach took place.

Summary

In this chapter, you learned about Zendesk's security options in direct correlation with recommended security practices. You learned about preventive security measures as well as the importance of ongoing checks. We reviewed Zendesk's SSO options and looked at two more complex examples (JWT and SAML).

I do encourage you to create a security document, outlining not only your standard security settings, but also your ongoing security tasks for you and your team. I also encourage you to talk to your company's web developers about the best SSO option for you.

In our next chapter, you will learn about a few different ways to troubleshoot your Zendesk setup.

9

Troubleshooting Zendesk

Almost every time we customize a system, it becomes more complex and therefore prone to error. When it comes to Zendesk, troubleshooting mainly consists of identifying the cause of unexpected behavior or dealing with overall performance issues.

While Zendesk's own support is great and will, most likely, locate and fix the problem for you, troubleshooting Zendesk yourself can save time and further your understanding of your setup.

In this chapter, we will highlight some of the most common issues and troubleshooting techniques. You will learn how to isolate problems and fix them accordingly.

The following topics will be covered in this chapter:

- Troubleshooting performance issues
- Troubleshooting business rules
- Troubleshooting Zendesk apps
- Zendesk support

Troubleshooting performance issues

We've all had this problem before. We need to get work done, but everything is way slower than usual. The occasional slowdown in performance plagues any online service available.

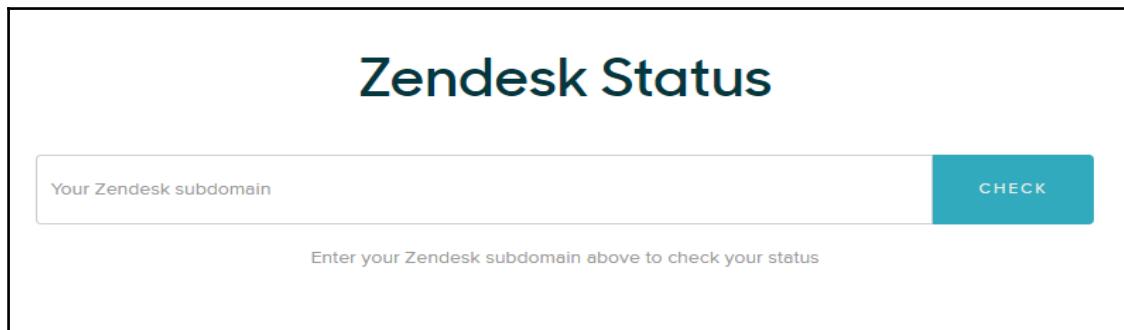
But why? Let's take a look at the possible causes and what we can do about them.

Zendesk status

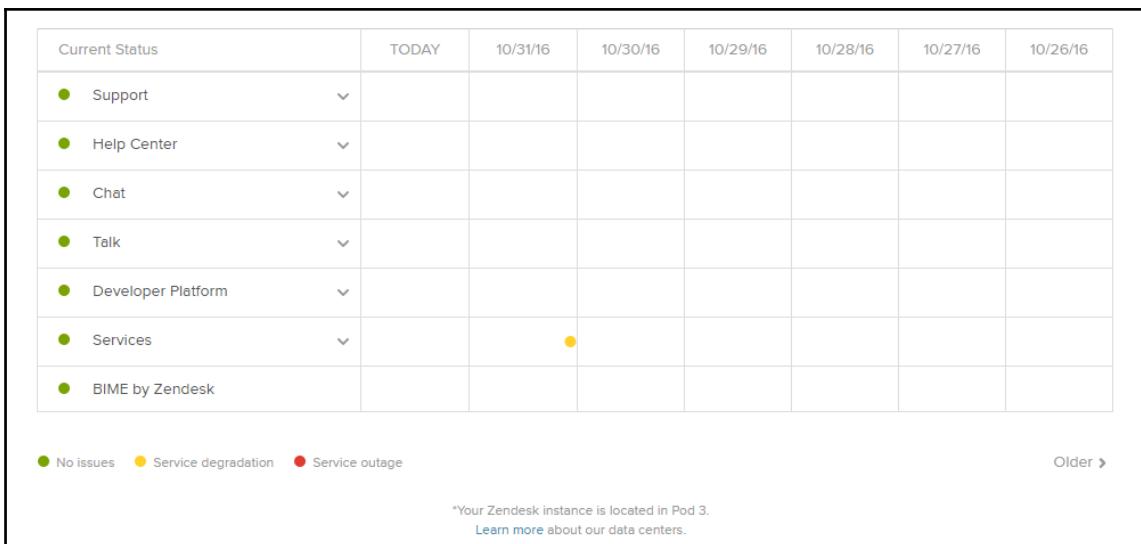
Before we start investigating our system and/or annoying our network administrators, we should check whether Zendesk is aware of any issues on their side.

We can do that by visiting the following website: <https://status.zendesk.com/>.

Zendesk has many data centers all over the world. This is why we will have to enter our subdomain in order to review the status of our “Pod”:



As soon as we have provided our subdomain, the website will display the current and past status for up to one week:

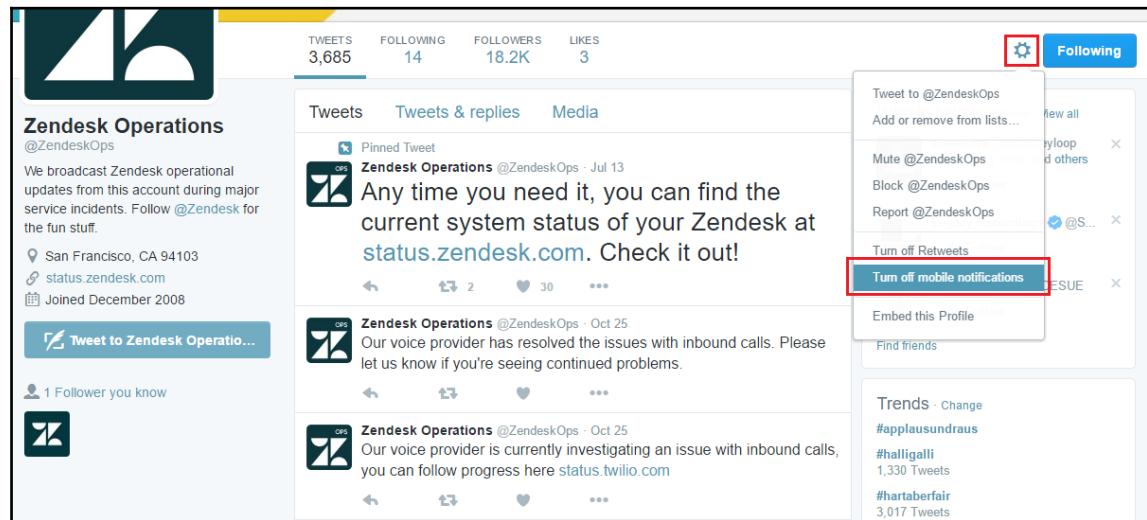


While the status report is divided into seven easy to understand categories, we can dig deeper by clicking on them:

Current Status	TODAY	10/31/16
● Support	^	
● Ticketing		
● Inbound Email Processing		
● Outbound Email Processing		
● Views		
● Triggers & Automations		
● SLAs		
● Mobile Apps		
● Support SDK		
● Web Widget		
● Misc		

We can also stay up to date about major service incidents by following the Twitter account Zendesk Operations at <https://twitter.com/zendeskops>.

Switching on mobile notifications allows us to stay one step ahead and become aware of performance issues before they have affected one of our users:



General browser issues

Often enough, our performance issues have nothing to do with Zendesk's servers and we can deal with them by troubleshooting our browser. If, for instance, one of our agents is reporting performance issues, but others (who are in the same network) do not encounter the same problem, we may want to ask the agent to restart the system first. If the issue persists, we can try troubleshooting the browser.

Disabling extensions/plugins

Ask your agent to review the browser's enabled extensions and plugins. If in doubt, disable all the extensions and plugins. If this fixes the performance issues, enable them one by one to isolate the cause.

Clearing browsing data/cache

Also, have your agents clear their browsing data. Clearing the cache gets rid of locale files, which may be outdated and therefore cause display and performance issues.

Traceroute

If nothing helped, we may want to report our performance issues to Zendesk, which in return might ask us to provide some additional information.

To save time, it makes sense to prevent any requests from their side and provide the necessary information with our initial contact.

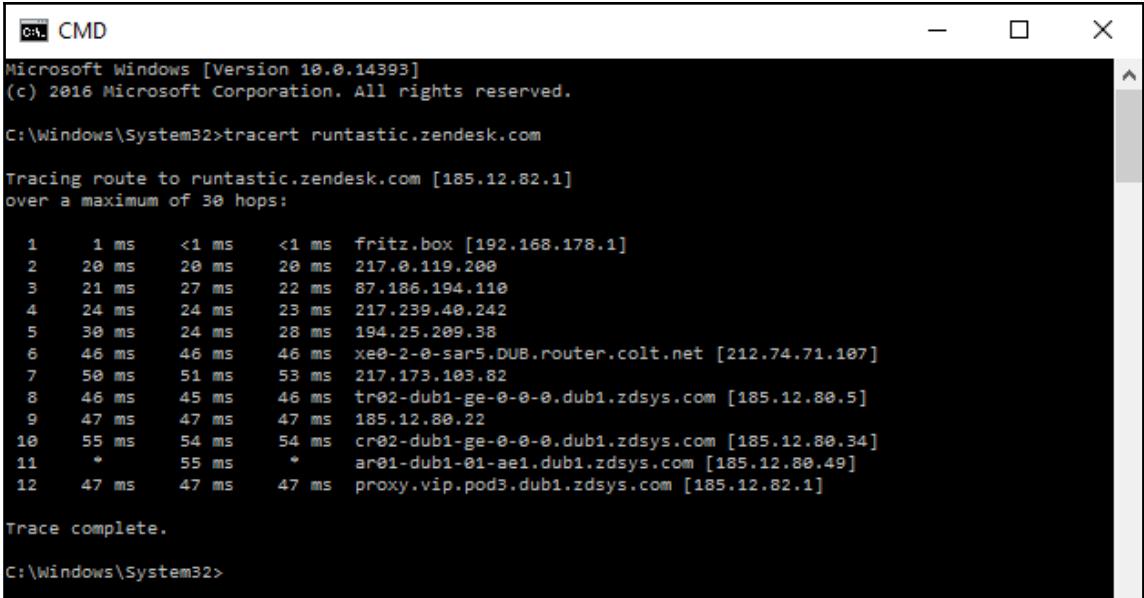
So how can we help Zendesk to investigate our report?

We can use a tool called Traceroute, which visualizes the path between us and the web server that you are trying to reach.

In order to use Traceroute, simply open the Command Prompt and enter the following command:

```
tracert yoursubdomain.zendesk.com
```

As soon as the tool finishes tracing the steps, it will put out **Trace complete**:



The screenshot shows a Windows Command Prompt window titled "CMD". The window displays the output of the "tracert" command. The output shows the traceroute path from the local machine to the Zendesk server "runtastic.zendesk.com". The path consists of 12 hops, with the final hop being the destination server. The output includes the hop number, round-trip time in milliseconds, and the IP address or name of each router or server along the path. The command prompt also shows the completion message "Trace complete." and the prompt "C:\Windows\System32>".

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\System32>tracert runtastic.zendesk.com

Tracing route to runtastic.zendesk.com [185.12.82.1]
over a maximum of 30 hops:

 1    1 ms    <1 ms    <1 ms  fritz.box [192.168.178.1]
 2    20 ms   20 ms    20 ms  217.0.119.200
 3    21 ms   27 ms    22 ms  87.186.194.110
 4    24 ms   24 ms    23 ms  217.239.40.242
 5    30 ms   24 ms    28 ms  194.25.209.38
 6    46 ms   46 ms    46 ms  xe0-2-0-sar5.DUB.router.colt.net [212.74.71.107]
 7    50 ms   51 ms    53 ms  217.173.103.82
 8    46 ms   45 ms    46 ms  tr02-dub1-ge-0-0-0.dub1.zdsys.com [185.12.80.5]
 9    47 ms   47 ms    47 ms  185.12.80.22
10    55 ms   54 ms    54 ms  cr02-dub1-ge-0-0-0.dub1.zdsys.com [185.12.80.34]
11    *      55 ms    *      ar01-dub1-01-ae1.dub1.zdsys.com [185.12.80.49]
12    47 ms   47 ms    47 ms  proxy.vip.pod3.dub1.zdsys.com [185.12.82.1]

Trace complete.

C:\Windows\System32>
```

We can then proceed by copying the results straight into our e-mail to support@zendesk.com.

Troubleshooting business rules

Having to troubleshoot business rules is one of the most common scenarios when working with Zendesk.

How so?

With each new trigger and automation, we undoubtedly add to the complexity of our setup. An older trigger, for instance, may malfunction because we did not consider adjusting it after deleting a ticket field. The possibilities are endless.

At the same time, we may have created a bunch of new SLAs, which simply do not seem to work as expected. So, let's take a look at our options.

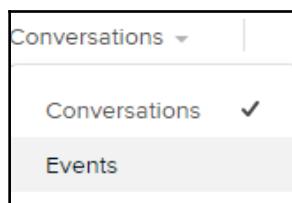
Triggers and automations

In most cases, trigger- or automation-related issues are reported in direct connection with a ticket. In other words, an agent finds a ticket and does not quite understand why it behaved the way it did.

Ticket events

In cases like the one mentioned earlier, we can review the ticket's history by looking at every event that took place since the ticket has been created.

In order to do so, simply click on **Conversations** on top of the ticket description and select **Events** from the context menu.



As a result, we can see every event that took place during our conversation with the customer.

Events are displayed just like this:



Zendesk will list each ticket property that has changed as well as its current and previous values.

If a property changed due to a trigger or automation, Zendesk will list this information as well:



Looking at the preceding example, we can deduct the following:

The **German tickets to Group DE** trigger changed the ticket's **Group** property to **Support DE**. Since the property did not have any other value assigned beforehand, Zendesk does not list the previous value at all.

Let's assume this is an unexpected behavior and the ticket should not have been assigned to the **Support DE** group, but **Support EN**, the English support group.

In this case, we should review two things:

- The **German tickets to Group DE** trigger, which leads to the property change
- The ticket's properties before this event

So let's take a look at the trigger:

Trigger title

German tickets to Group DE

Meet **all** of the following conditions:

Requester: Language Is German

Ticket: Group Is not Support DE

Ticket: Is... Created

Add condition

Meet **any** of the following conditions:

-- Click to select condition. --

Add condition

Perform these actions:

Ticket: Group Support DE

Add action

The screenshot shows the configuration of a Zendesk trigger titled "German tickets to Group DE". It starts by defining conditions that must be met simultaneously ("all"). These include checking the requester's language is "German" and ensuring the ticket is not assigned to the "Support DE" group. It also includes a condition where the ticket was created. Below this, there is a section for defining conditions that can be met individually ("any"), which currently has a placeholder message. Following this, there is a section for "actions" that will be performed when the trigger fires, specifically setting the ticket's group to "Support DE".

So what does the trigger do?

As soon as a ticket has been created, it will check the requester's language and make sure the ticket has not yet been assigned to the **Support DE** group. If all these conditions are met, the trigger will commence and set the ticket's group accordingly.

Our verdict is that the trigger seems fine.

Next, we will have to review the ticket's properties before the trigger fired, namely the requester's language, which seems to be the only property that may have caused the unexpected outcome.

We are lucky! It seems as if there is nothing wrong with our trigger. The user contacted us in German a few weeks before, which lead to Zendesk setting the user's language to German. His most recent ticket, however, was written in English, which left the agent believing that something is wrong with our setup.

The agent can simply adjust the group manually.

While this was a rather simple example, it will always boil down to these exact two steps:

1. Reviewing the business rule that changed the property.
2. Reviewing the ticket's properties (used in the rules conditions) before this event.

In some cases, we might have to go through more than one trigger and/or automation, and follow the audit all the way through until we find the initial cause of the problem.

Trigger order

In some cases, something as simple as the order of our triggers may cause expansive issues.

How so?

To understand this problem, let's take a closer look at the way how triggers fire:

- Every time a ticket is created or updated, Zendesk runs through all the triggers
- As soon as one trigger's conditions are met, it fires
- Zendesk will then start back at the top, running through all triggers, except the one that fired already

This is why triggers can affect each other and lead to unexpected results.

SLAs

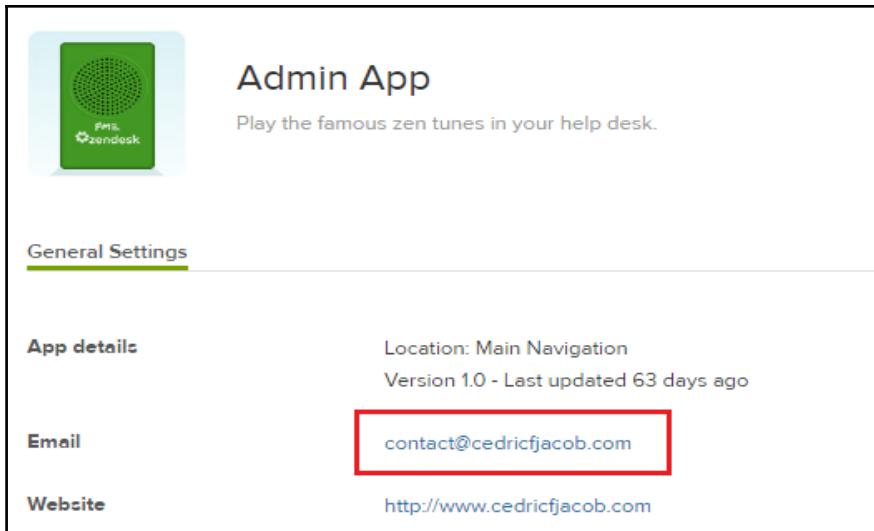
If one or more of your SLAs do not perform as expected, you may want to go through the following list of pointers:

- SLAs are only matched to a ticket when an event takes place (ticket created, ticket updated). If you create a new SLA, it will not be matched with the existing tickets until they are being updated.
- A ticket can only be matched to an SLA as long as the ticket priority has been set.
- If an agent creates a ticket on the behalf of the end user, the First Reply Time metric will not work.

Troubleshooting Zendesk apps

Troubleshooting Zendesk Apps is definitely a job for our developers, or more likely the developer of the app itself.

If the issue persists after reinstalling the app, we can find the developer's e-mail address in our app's general settings:



The screenshot shows the 'Admin App' settings page. At the top, there is a green square icon with a white 'zendesk' logo. Below it, the app name 'Admin App' is displayed in bold, followed by the subtitle 'Play the famous zen tunes in your help desk.' Under the heading 'General Settings', there are two sections: 'App details' and 'Email'. The 'App details' section shows 'Location: Main Navigation' and 'Version 1.0 - Last updated 63 days ago'. The 'Email' section shows the developer's email address, 'contact@cedricfjacob.com', which is highlighted with a red rectangular box. Below the 'Email' section, there is a 'Website' section with the URL 'http://www.cedricfjacob.com'.

However, there is something we can do to help the developer understand the exact issue.

All modern web browsers come with developer tools. These tools can be used for a range of things, such as inspecting a website's HTML structure.

How does one open those tools?

Opening these tools varies depending on the browser and your OS. Here are some examples:

- **Chrome:** *Ctrl + Shift + I*
- **Internet Explorer:** *F12*
- **Mac OS X:** *⌘ + ⌘ + I*

The developer tools will look something like this:



We are after the **Console**, which uses log diagnostic information during web development.

If our app does not work as expected, it is likely that we will find an error message in the console. When in doubt, we can simply copy the whole output and forward it to the developer.

Contacting Zendesk support

Last but not least, we always have the option to contact Zendesk's own support team.

We can contact the support by creating a ticket within their Help Center: <https://support.zendesk.com/hc/en-us/requests/new>.

Or we can contact their support by sending an e-mail to support@zendesk.com.

Summary

In this chapter, you learned how to troubleshoot our Zendesk setup. We also looked at different ways that we can help third parties to understand our exact issue, which should result in faster support.

Over time, you will become better at spotting the cause of a problem within your setup. Do not become discouraged and never quash the idea of contacting Zendesk's support team.

In the next chapter, we will look at a range of tips and tricks that I have collected over my years of working with Zendesk.

10

Zendesk Tips and Tricks

When it comes to customizing Zendesk, experience is hard to beat. Admins might spend years working on the perfect setup and they keep learning along the way.

We can certainly try to catch up by learning as much about Zendesk as possible, and yet, we still miss one key ingredient—Time. Only when spending time with Zendesk, we will come up with our own unique ideas and solutions.

While this rings true, we can certainly jumpstart our “experience” by examining tips and tricks of the pros.

In this chapter, we will tap into that experience by looking at my compilation of tips and tricks for Zendesk.

This chapter will cover tips and tricks for the following topics:

- Business rules
- Roles and views
- Reporting
- Zendesk apps

Business rules

We already know how to use business rules for their “intended purpose”. With time, you may find yourself using business rules to achieve all kinds of tasks.

The following ideas may spark your inspiration and help you to come up with your own ideas to utilize them further.

Notifying the team leader about a ticket with a high amount of replies

When it comes to good support, the amount of replies required to successfully solve a ticket can be a great indication of the agent's response quality.

Having to look out for tickets with an unusual number of agent replies means having to actively seek them out. No matter how much time one would spend on such a task, automating this process makes sense.

In order to do so, we only need to create a simple trigger:

The screenshot shows the configuration of a Zendesk Business Rules trigger. The title is "Notify team leader about tickets with high amount of replies". The condition is set to "Meet all of the following conditions". There are three conditions listed: 1) Ticket: Is... Updated - (with a minus sign icon). 2) Ticket: Agent replies Is 5 - (with a minus sign icon). 3) Other: Current user Is (agent) - (with a minus sign icon). A green plus sign icon at the bottom right allows adding more conditions.

Trigger title

Notify team leader about tickets with high amount of replies

Meet **all** of the following conditions:

Ticket: Is... Updated -

Ticket: Agent replies Is 5 -

Other: Current user Is (agent) -

Add condition +

The trigger will fire as soon as an agent updates a ticket with their fifth reply:

The screenshot shows the configuration of a Zendesk automation trigger. It is set to 'Perform these actions' and targets 'Notifications: Email user' sent to 'Cedric Jacob'. The trigger fires when a ticket is updated and an agent replies. The email subject is set to `{{ticket.id}} - already 5 agent replies` and the body contains `{{ticket.link}}` and `{{ticket.comments_formatted}}`. A red minus sign icon is present in the top right corner of the action panel.

Perform these actions:

Notifications: Email user ▾ Cedric Jacob ▾ -

Email subject:

`{{ticket.id}} - already 5 agent replies`

Email body:

`{{ticket.link}}`
 `{{ticket.comments_formatted}}`

[View available placeholders »](#)

The trigger then notifies the team leader via e-mail providing the ticket's content and a direct link for further review.

We can use this concept to notify team-leaders or agents about all sorts of situations. Some examples are as follows:

- An end user who has not received a timely response
- A ticket that has not been assigned to an agent within the desired timeframe
- A ticket that contains words or phrases that might indicate a high level of urgency

Using tags to disable or enable business rules for tickets

While automating processes in Zendesk can save us a lot of time. Sometimes, upon reviewing a ticket, we realize that a certain automation should not be applied to this ticket after all; or vice versa, we want to force an automation to take action even though the ticket does not meet the predefined definitions within the business rule.

With a simple trick, we can force Zendesk's hand. Let's have a look at two examples.

Closing a ticket via a tag

For agents, closing tickets manually in Zendesk is not an option.

Instead, we only may choose to solve a ticket and wait for the designated automation to close the ticket within the time frame specified. But what if we want to close the ticket right away in order to prevent any more replies by the end user?

To achieve this, again, we only need to create a simple trigger:

Trigger title

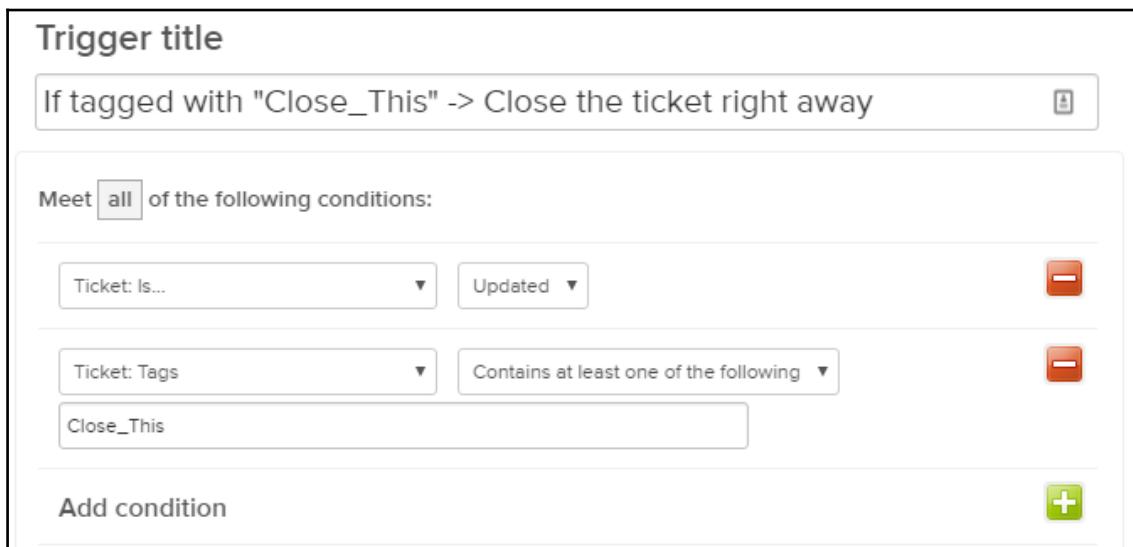
If tagged with "Close_This" -> Close the ticket right away

Meet **all** of the following conditions:

Ticket: Is... Updated

Ticket: Tags Contains at least one of the following
Close_This

Add condition

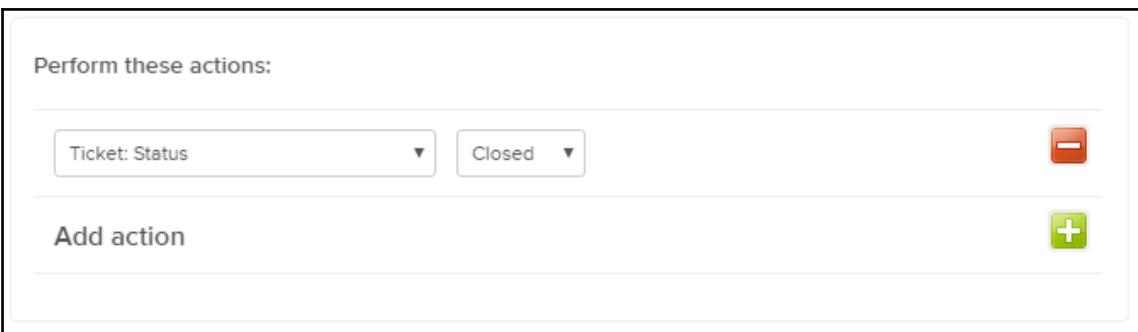


The trigger will fire as soon as an agent updates the ticket adding the **Close_This** tag:

Perform these actions:

Ticket: Status Closed

Add action

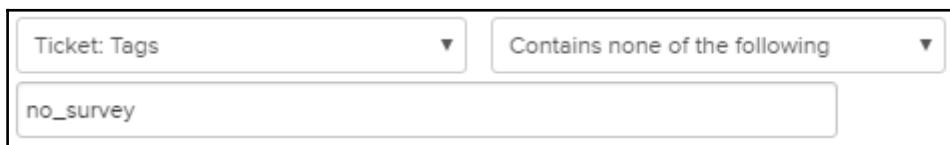


It will then commence and set the ticket's status to Closed. The same concept can be applied to force any trigger to take action right away.

Disabling the satisfaction survey via a tag

Sending satisfaction surveys to our users enables us to measure our performance and take action to improve if necessary. But what if an end user decides to misuse this survey? In some cases, we may want to disable this survey for a specific user.

Again, we can make use of a business rule to achieve the desired outcome. In this case, we will alter the existing automation that is designed to send the survey to our users by adding the following condition:



This will prevent the automation from firing as long as the **no_survey** tag is present in the ticket.

If necessary, we can use this exact technique in order to prevent any number of business rules to take action.

For example, you may want to set up specific tags to prevent the following:

- An end user from receiving e-mail notifications
- A ticket from being closed after a specific time frame
- An automatic reply from being sent out to a specific user

Removing tags from follow-up tickets

Closing a ticket will prevent end users and agents from altering the ticket again. In some cases, an end user or agent might choose to create a follow-up ticket. While the follow-up ticket receives its own ticket ID, it will contain a reference to the original ticket and the data stored in ticket-fields will be pulled over as well.

This makes sense in most cases. In some cases, it may lead to problems.

For instance, previously, we created a trigger that closes any ticket tagged with "Close_This". If this tag were to be automatically added to any follow-up ticket, any attempt to create such a follow-up ticket would just lead to the ticket being closed again automatically.

To fix this, we can choose to remove specific tags from follow-up tickets or to remove them altogether by creating a simple trigger, such as the one shown in the following screenshot:

Trigger title

Follow up tickets -> Remove tags

Meet **all** of the following conditions:

Ticket: Is...	Created	-	
Ticket: Channel	Is	Closed ticket	-

Add condition +

The trigger fires as soon as a ticket is being created from a closed ticket, meaning the ticket is a follow-up ticket:

Perform these actions:

Ticket: Set tags	-

This action will remove existing tags and replace with the tags you specify (use spaces to separate individual tags)

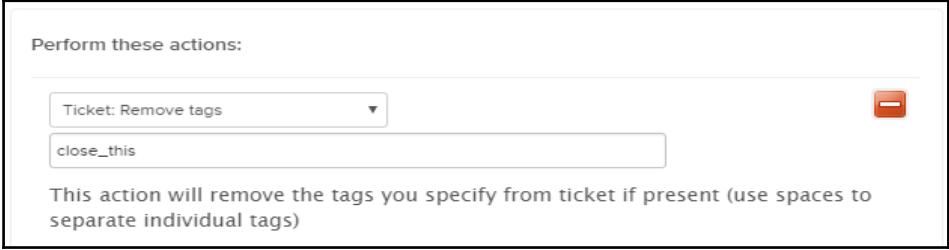
It will then go ahead and remove all existing tags. Sometimes, we only want to remove tags that might cause problems in our setup. In this case, our action would look something like this:

Perform these actions:

Ticket: Remove tags -

close_this

This action will remove the tags you specify from ticket if present (use spaces to separate individual tags)



Using business rules to detect and fix user errors

When working with complex systems, eventually we will come across some user errors. In some cases, if we are aware of these errors, we can set up business rules to monitor and auto-correct tickets accordingly.

Let's assume we have created specific guidelines for our agents, outlining how to deal with incident tickets. They state that the status of incident tickets should always be set to *On-hold*.

In some cases, however, agents accidentally set the status to *pending*.

By creating a simple trigger, we can not only avoid similar mistakes in the future, but educate our staff at the same time:

Trigger title

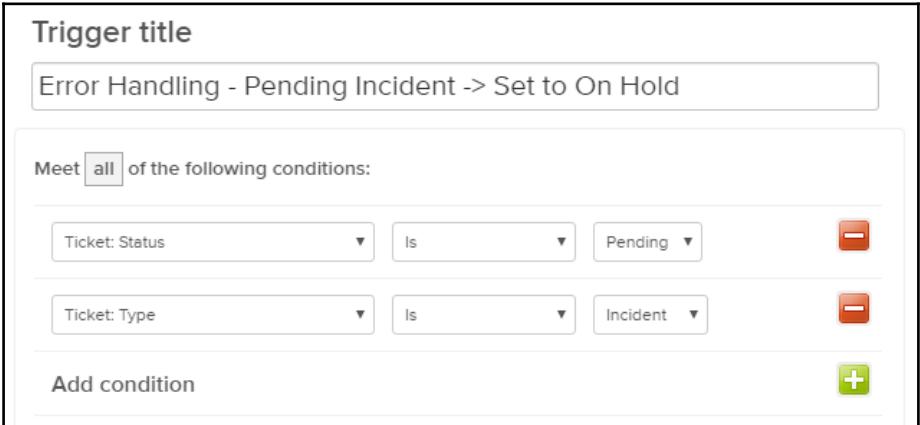
Error Handling - Pending Incident -> Set to On Hold

Meet all of the following conditions:

Ticket: Status Is Pending -

Ticket: Type Is Incident -

Add condition +



This trigger fires as soon as an **Incident** ticket's status is set to **Pending**:

Perform these actions:

Ticket: Status

Notifications: Email user

Email subject:
Error corrected - {{ticket.id}} was set to ON HOLD

Email body:
{ {ticket.link}}
Please note: All incidents should be set to ON HOLD.

[View available placeholders »](#)

It then automatically corrects the ticket's status and sends an e-mail notification to the agent.

Adding comments to tickets via a URL target

You may have noticed already that Zendesk's business rules do not allow us to automatically add any comments to a ticket.

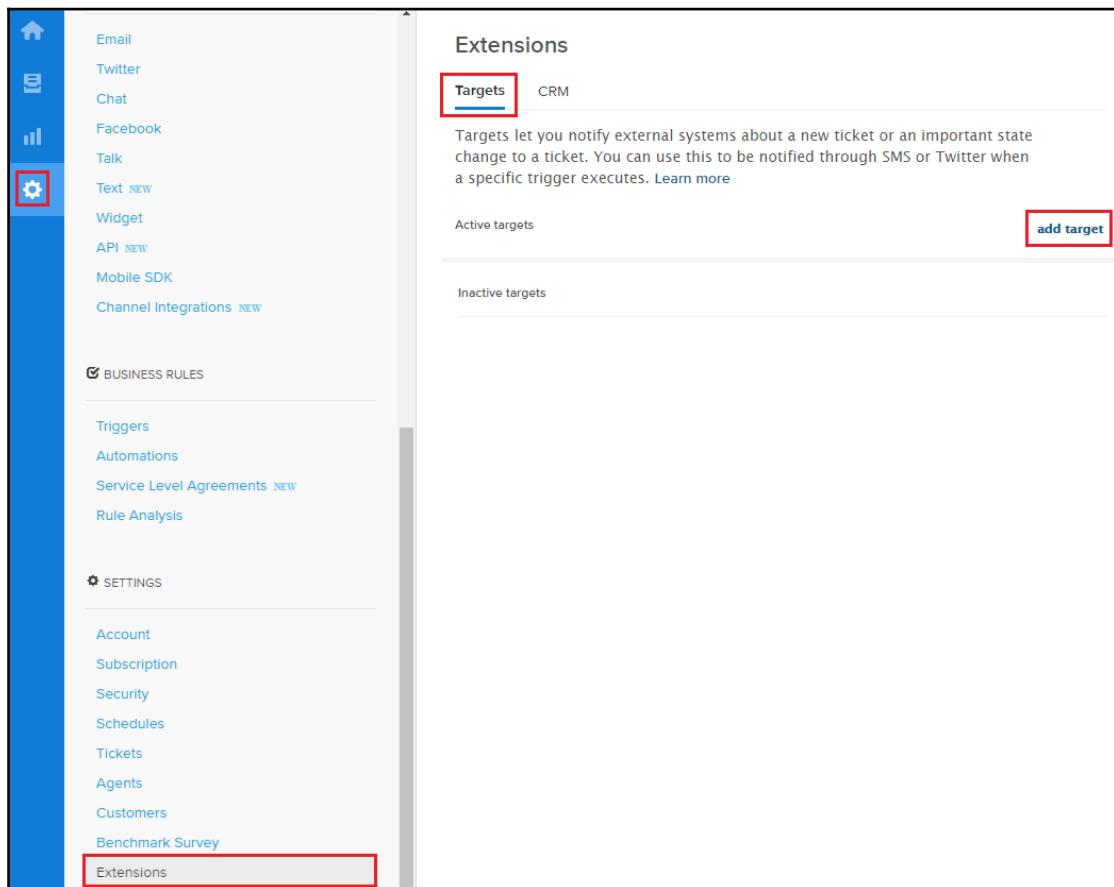
In some cases, however, we may want to do exactly that. For instance, having a trigger automatically add important information for the agent as a private comment would be a great feature.

So how can we accomplish that feat?

The answer lays in creating a URL target that calls the Zendesk API to add the comment.

Let's start by adding the target:

1. Click on the Admin icon (gear symbol) located in Zendesk's sidebar.
2. Click on **Extensions** located under **SETTINGS** within the admin menu.
3. In order to create a new target, navigate to the **Targets** tab and click on **add target**:



Zendesk will list different types of targets. Locate the **URL target** and click on it:



Name the target, enter the **Url**, choose the **Method**, and enter **Attribute Name**:

URL target

Title	<input type="text" value="Add Public Comment to Ticket"/> Edit
Url	<input type="text" value="https://examplecomp.zendesk.com/api/v2/tickets/{{ticket}}"/> The target URL, including protocol (https or http is OK) and path. Valid examples: <ul style="list-style-type: none">• http://somedomain/a/path• http://somedomain/a/path?source=zendesk• http://somedomain/a/path?id={{ticket.external_id}}&status={{ticket.status}}
Method	<input style="width: 100px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px; margin-bottom: 5px;" type="button" value="PUT"/>
Attribute Name	<input type="text" value="ticket[comment][body]"/> The name of the message attribute If the name of the message attribute is value, for example, your Notify Target action message will be appended to the target URL as <i>http://somedomain/a/path?value=message+with+placeholders+evaluated</i>
Basic Authentication	Add credentials if the target needs username/password authentication. Leave blank if credentials are not required.
Username	<input type="text" value="cedric.jacob@examplecomp.com"/>
Password	<input type="password" value="....."/>

In this case, we are creating a target allowing us to add a public comment to a ticket:

1. For the URL, enter the following and replace `yourdomain` accordingly:

```
https://yourdomain.zendesk.com/api/v2/tickets/{{ticket.id}}.json?ticket[comment][public]=true
```

2. Choose **PUT** from the **Method** dropdown.
3. Enter the following as **Attribute Name**: `ticket[comment][body]`.
4. Make sure to enter your credentials and save the target.

Now we can use the target in our business rules. Let's create a quick test trigger:

The screenshot shows the 'Add public comment to ticket' configuration screen. At the top, it says 'Meet all of the following conditions:' with a dropdown set to 'all'. Below this are three condition rows, each with a red minus button on the right.

- Row 1:** 'Ticket: Is...' dropdown, 'Updated' dropdown, and a red minus button.
- Row 2:** 'Ticket: Tags' dropdown, 'Contains at least one of the following' dropdown, and a red minus button. The input field contains 'test_target'.
- Row 3:** 'Ticket: Tags' dropdown, 'Contains none of the following' dropdown, and a red minus button. The input field contains 'test_target_done'.

Our test trigger fires when updating a ticket with the **test_target** tag. In order to prevent the trigger from looping, it will not fire when the **test_target_done** tag is present:

The screenshot shows the Zendesk trigger configuration interface. At the top, there's a section titled "Perform these actions:" containing two dropdown menus: "Notifications: Notify target" and "Add Public Comment to Ticket". Each dropdown has a red minus sign icon to its right. Below this is a "Message:" field containing the text "Have a great day!". Underneath the message field is a link "View available placeholders »". Further down, there's another set of action fields: "Ticket: Add tags" with a dropdown and a minus sign, and a single input field containing the tag "test_target_done".

The trigger will then notify the target and add the **test_target_done** tag to prevent the trigger from firing again.



Note: Zendesk does not officially support this work-around. If Zendesk was to implement changes affecting the functionality of this hack, you will have to find a new way to achieve similar results. This example is meant to illustrate how existing functionality can be repurposed.

Logging ticket events via an e-mail target

Keeping event logs can be very helpful when it comes to debugging business rules. Some prefer keeping additional logs outside Zendesk.

We can achieve that by sending e-mail notifications to a designated e-mail address whenever a trigger or automation fires. To avoid having to use an agent's e-mail address, we can set up an e-mail target:

The screenshot shows the 'Email target' configuration page. It includes fields for 'Title' (Ticket Event Logs), 'Email Address' (logs@examplecomp.com), 'Subject' (LOG Email | Ticket ID: {{ticket.id}}), and a note about using plain text or placeholders in the subject. Below the subject field is a list of two bullet points: 'Please look at this.' and 'This is ticket {{ticket.id}} from {{ticket.account}}'. At the bottom are 'Test Target' and 'Submit' buttons.

Email target	
Title	Ticket Event Logs
Email Address	logs@examplecomp.com
The destination email address	
Subject	LOG Email Ticket ID: {{ticket.id}}
The subject of the email	
You can use plain text or placeholders in the subject:	
<ul style="list-style-type: none">• Please look at this.• This is ticket {{ticket.id}} from {{ticket.account}}	
Test Target	Submit

After submitting the new target, we can start using it in our business rules:

The screenshot shows a 'Business Rules' configuration screen with a 'Message:' section. It contains placeholder text for ticket details: 'Title: {{ticket.title}}', 'ID: {{ticket.id}}', 'Link: {{ticket.link}}', 'Status: {{ticket.status}}', 'Priority: {{ticket.priority}}', and 'Tags: {{ticket.tags}}'.

Notifications: Notify target	Ticket Event Logs
Message:	
Title: {{ticket.title}} ID: {{ticket.id}} Link: {{ticket.link}} Status: {{ticket.status}} Priority: {{ticket.priority}} Tags: {{ticket.tags}}	

Make sure to include any information that might be useful later on.

Roles and views

Although roles and views are very basic components of Zendesk, it does not mean that we do not learn about new ways to approach them along the way.

Creating a vacation view

At times, one or more of our agents will enjoy some deserved off-time. There are a few ways to handle situations like this.

Creating a vacation view and displaying all their open tickets is one of the easier and most practical solutions:

The screenshot shows the 'View title' field containing 'Vacation View'. Below it, under 'Meet all of the following conditions:', there is a single condition: 'Ticket: Status Is Open'. A red minus sign button is available to remove this condition. An 'Add condition' button with a green plus sign is present. Under 'Meet any of the following conditions:', there is another condition: 'Ticket: Assignee Is Daniel D'. Similar to the first section, a red minus sign button is available to remove this condition, and an 'Add condition' button with a green plus sign is present. At the bottom right, a button says 'Preview match for the conditions above'.

Adding the **Ticket: Assignee** condition to the **any** section allows us to add more than one agent at a time when needed.

Creating a history view

Being able to review previously updated tickets can be very helpful; especially, if we want to look back at a specific ticket that we dealt with before.

The easiest way to achieve this is by creating a view:

The screenshot shows the 'View title' configuration for a 'History' view. The title 'History' is entered in the main field. Below it, the condition 'Meet all of the following conditions:' is set to 'Ticket: Assignee Is (current user)'. There is also an 'Add condition' button and a delete button.

We will need to make sure that the **Latest update by assignee** column is included in our table view:

The screenshot shows the 'Columns included in table' configuration, where the 'Latest update by assignee' column is selected.

We should also order the tickets by **Latest update by assignee** and select **Descending**:

The screenshot shows the 'Order by' configuration, where 'Latest update by assignee' is selected and the sorting direction is set to 'Descending'.

There we go. We can now view previously updated tickets by opening the **History** view.

But what happens if we touched a ticket and then assigned it to another agent? How could we make sure this ticket is included in our **History** view?

We could, for instance, create a trigger that tags every ticket that we touch. The tag would have to be individual such as *updated_by_cedric*. To make this work, we would have to create a private view displaying tickets with this exact tag instead of only checking for the current assignee.

While this works, looking back at a ticket might be confusing. What did we do with this ticket? Why is it in our history view? To understand, we would have to review the ticket's event log.

Can we find a better solution?

In addition to adding the tag, we could use a URL target and a trigger to add a private comment to a ticket whenever we update it (refer to the *Adding Comments to Tickets via URL Target* section). This private comment would say something, such as “Updated by Cedric”, making it easier to locate the exact moment when we touched the ticket.

Creating limited custom roles for new staff members

In Chapter 2, *Agent Roles, Groups, Organizations, and User Tags*, you learned about custom roles and how to create them. You also learned to limit each agent's permissions for security reasons.

Another reason to limit permissions are new staff members. During their training, staff members will, most likely, go through different learning stages. Slowly adding permissions when they become necessary can help to reduce the chance of human error.

Think about your staff's learning stages and create separate roles for each stage. You may also want to create different groups limiting the views available to new agents.

Reporting

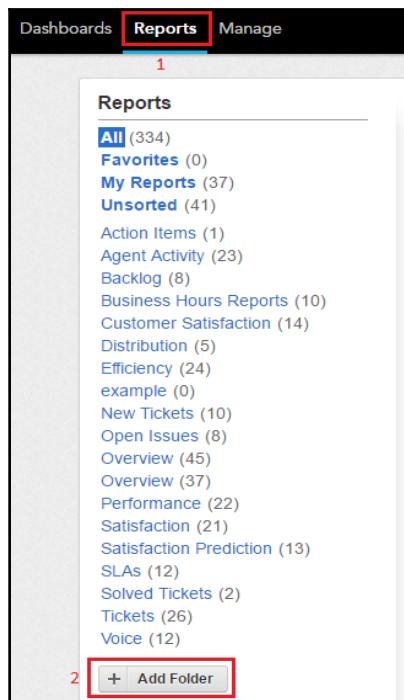
GoodData is a very complex and multifaceted tool. Hardly a session goes by without picking up on new little details that allow us to optimize and improve our reports. Let's go through some of my favorite GoodData tricks.

Report tags and folders in GoodData

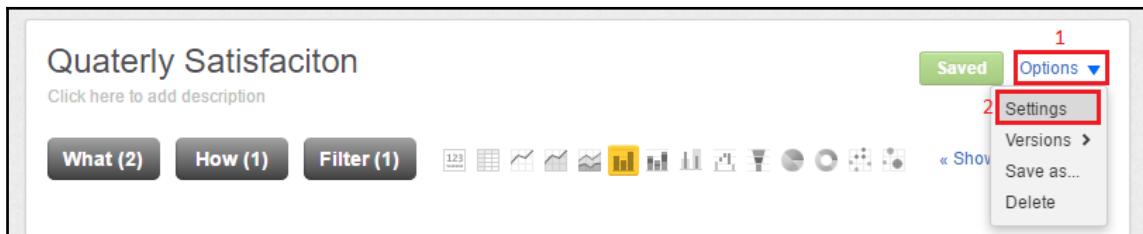
During the course of planning and executing the perfect GoodData dashboard, it is very likely that we will create a substantial amount of reports. In many cases, we will keep creating different reports and extending our dashboards constantly.

Making sure to pick clear and unique names can only help so much. At some point, we will lose track of all the different reports and their names. Luckily, GoodData allows us to create a folder structure as well as to tag our reports.

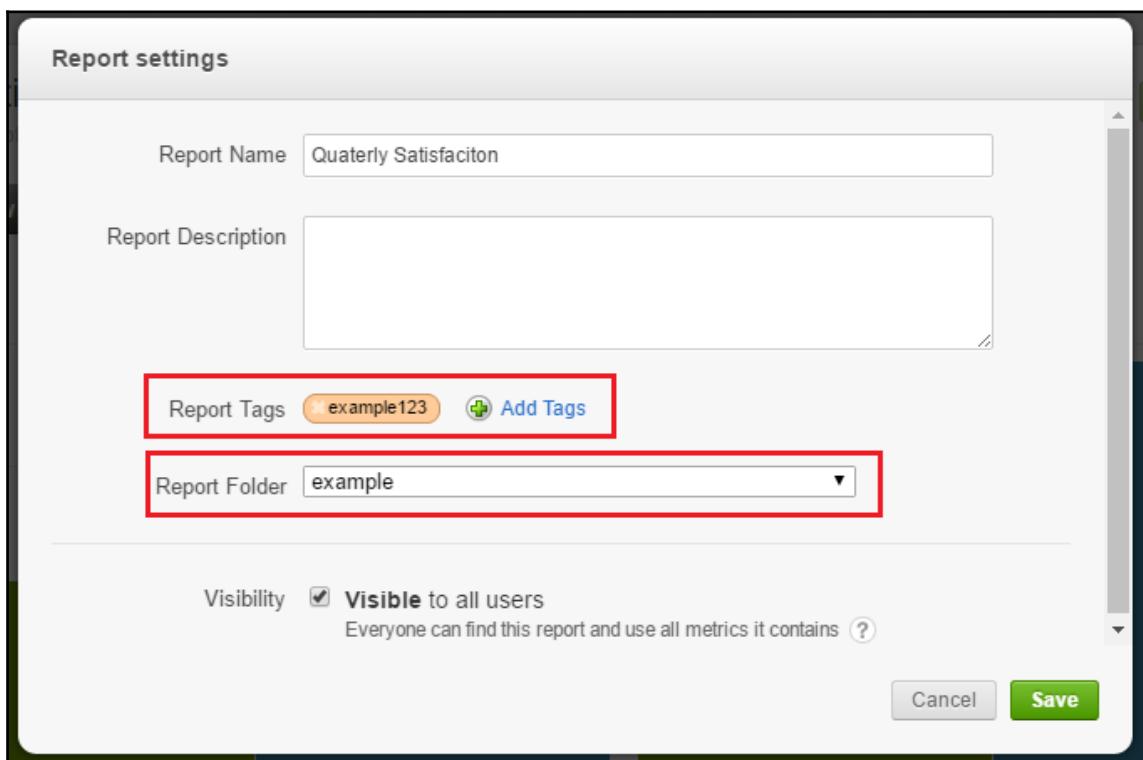
In order to create a folder, navigate to the **Reports** tab and click on **Add Folder**:



Once a folder has been created, we can assign reports by editing the report and navigating to its **Settings**:



Once in the settings, we can also add as many tags as we like, making it easier to locate the report again later on:

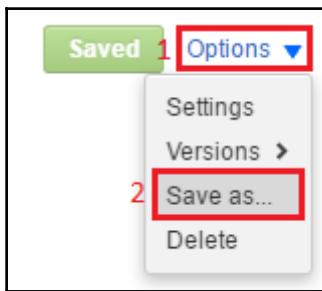


Investing some initial time and creating a folder structure before creating too many reports can go a long way when working with GoodData. We will save time in the long run and keep our GoodData environment clean.

Duplicating reports in GoodData

Often enough, we will have to create similar reports and only change one attribute. Rather than starting from scratch, we can duplicate the report and make the necessary changes to the copy.

When editing a report, simply click on **Options** followed by **Save as...** and enter a different name:



Excluding specific tickets from your GoodData reports

There are many reasons why you would want to exclude certain tickets from your reports.

For instance, your software might push feedback tickets to your support environment. These tickets are then automatically closed and archived for later review. These tickets can easily impact your performance reviews to the point that they just do not make sense anymore.

One solution would lay in excluding all unwanted tickets from every report. This would take a lot of time and require a lot of upkeep as we might have to update every report again and again whenever we need to exclude a bunch of tickets.

A better solution would lay in adapting the **# Tickets** core metric.

In order to access a metric, simply follow these steps:

1. Click on **Manage** in GoodData's navigation bar.
2. Click on **Metrics** located on the left-hand side.
3. Locate and click on the **# Tickets** metric:

The screenshot shows the GoodData interface with the 'Manage' tab selected in the top navigation bar. On the left, there is a sidebar with tabs for 'Data' (selected) and 'Emailing Dashboards'. Under 'Data', the 'Metrics' tab is highlighted with a red box. In the main area, there is a table titled 'Metrics' with columns for 'Title', 'Time', and 'Author'. The first row, which includes the '# Tickets' metric, is also highlighted with a red box. The table contains the following data:

Title	Time	Author
# Tickets	Dec 14, 2016	Admin Zendesk prod
# Agent Stations	Oct 02, 2014	Admin Zendesk prod
# Assignee Stations	Jul 01, 2016	Admin Zendesk prod
# Backlog Tickets	Jul 01, 2016	Admin Zendesk prod

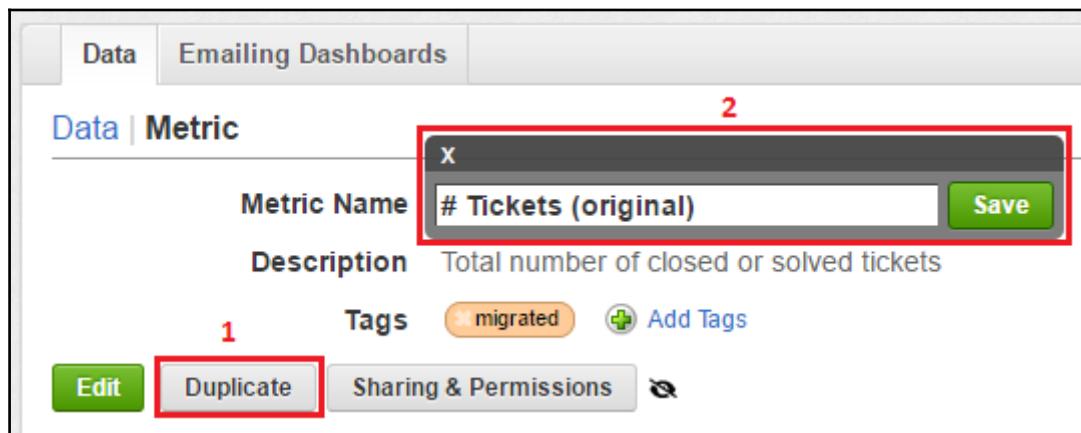
The following will pop up:

The screenshot shows the 'Metric' details view for the '# Tickets' metric. The top navigation bar has 'Data' selected. The metric name is '# Tickets' and its description is 'Total number of closed or solved tickets'. It has a 'migrated' tag and a 'Tags' button. Below the metric details, the MAQL query is shown:

```
SELECT # Tickets WHERE Ticket Status IN (Closed, Solved, Hold, Open, Pending, New)
```

Under 'Metric Format', the format is set to '#,##0' with an 'Edit' button. The 'Metric Comments' section has a 'Add Comment' button.

We could go ahead and edit the MAQL of this metric to exclude the tickets we do not want in our reports. A better way would be to create a **Duplicate** first and name it something such as **# Tickets (original)**:



Now that we have copied the metric and marked the copy as being the original (unchanged) version, we can go ahead and navigate back to the **# Tickets** metric and start editing the MAQL. This is what it should look like:

Make sure that you opened the right metric, not the copy!



The screenshot shows the Metric Editor interface. The title bar says 'Metric Editor'. The first section is 'Name your metric:' with the input field containing '# Tickets'. The second section is a code editor containing the following MAQL query:

```
SELECT # Tickets WHERE Ticket Status IN [Closed, Solved, Hold, Open, Pending, New]
```

Now we can use the previously created copy as the base of our new MAQL and exclude tickets with certain tags, as you learned in [Chapter 7, Advanced Reporting and Insights via GoodData](#):

The screenshot shows the Zendesk Metric Editor. At the top, it says "Metric Editor". Below that, there's a section labeled "Name your metric:" with a text input field containing "# Tickets". Underneath, there's a code editor area containing the following MAQL query:

```
SELECT IFNULL((SELECT # Tickets (original) WHERE Ticket Tag = feedback),0)
```

Sending agent-specific reports using variables

In [Chapter 7, Advanced Reporting and Insights via GoodData](#) you already learned how to e-mail dashboards to our agent.

But what if we do not want every agent to see the statistics of their peers? How can we limit them to only seeing their own statistics?

Well, we could go ahead and create one dashboard for every single agent. This way we would have to create a lot of duplicate reports and change the filter for each agent accordingly.

Luckily for us, there is a better way—**Variables**.

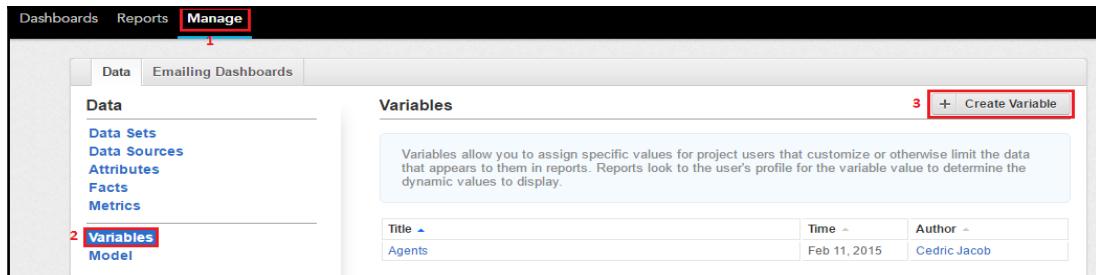
Variables allow you to assign specific values for project users that customize or otherwise limit the data that appears to them in reports. Reports look to the user's profile for the variable value to determine the dynamic values to display.

— GoodData

So how does this work?

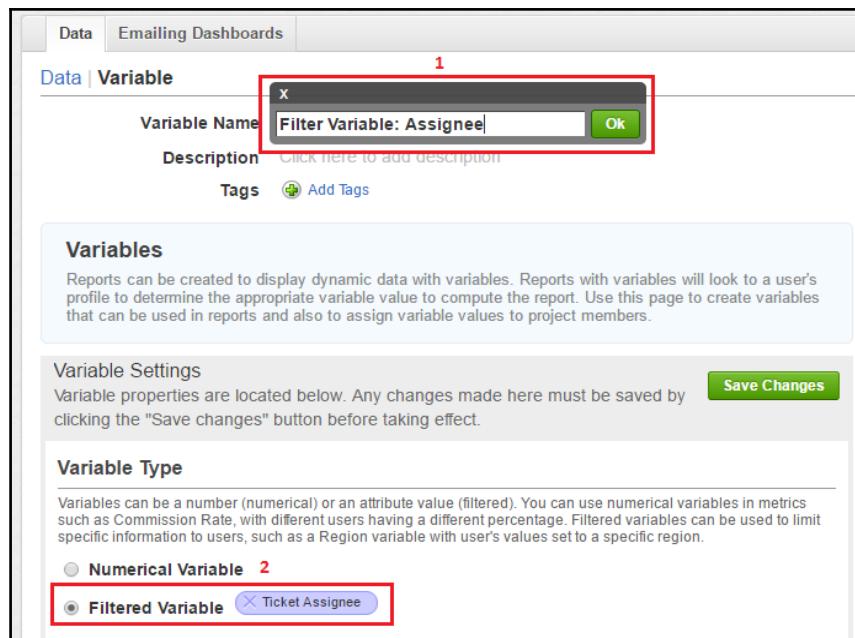
First, we will need to create a variable:

1. Click on **Manage** in GoodData's navigation bar.
2. Click on **Variables** located on the left-hand side.
3. Click on **Create Variable** on the right-hand side:



First, GoodData will ask us to name our variable. As we are planning to use this variable as an additional filter, let's go with **Filter Variable: Assignee**.

After naming the variable, we can choose between a **Numerical Variable** or a **Filtered Variable**. We will go for the latter and choose the **Ticket Assignee** attribute:

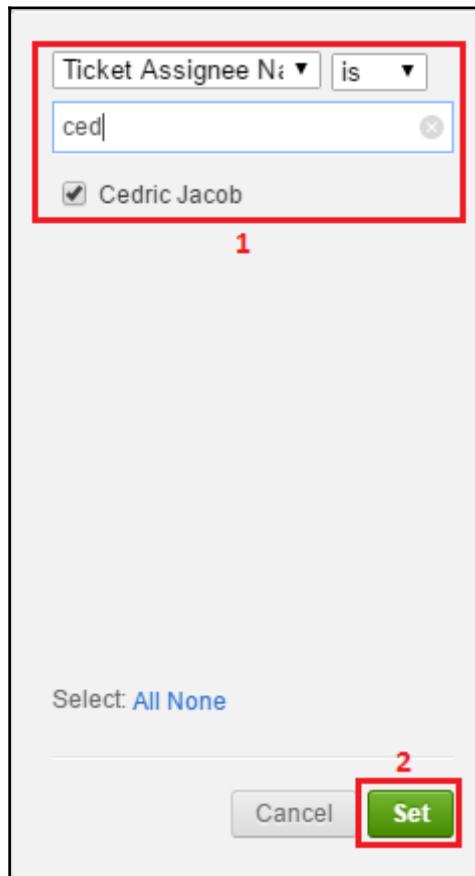


At the bottom of the page, GoodData lists all the project users. Here we can assign an individual value to the variable for each user.

To do so, simply click on **Choose** located next to each list entry:

Jacob, Cedric	Zendesk Editor	(uses default value)	All Choose
---------------	----------------	----------------------	------------

In our case, we want to assign the associated agent name for every user:



So what did we do here?

We created a variable with the **Ticket Assignee** attribute. We then went on and assigned a value for every project user that must ring true in order for the data to be displayed. In our case, the **Ticket Assignee** attribute must be the agent name of the viewer.

After saving our variable, we can go ahead and start using it in our reports by adding it as a filter:

1. When creating or editing a report, navigate to the **Filter** tab.
2. Click on **Variable Filter** at the bottom of the list:

Example Report

Click here to add description

What **How** **Filter**

Select one of the filter types below:

Select from a List of Values (including date ranges)
Pick values from a list. (Example: Year is 2006, 2007 or "Last 4 Qtrs")

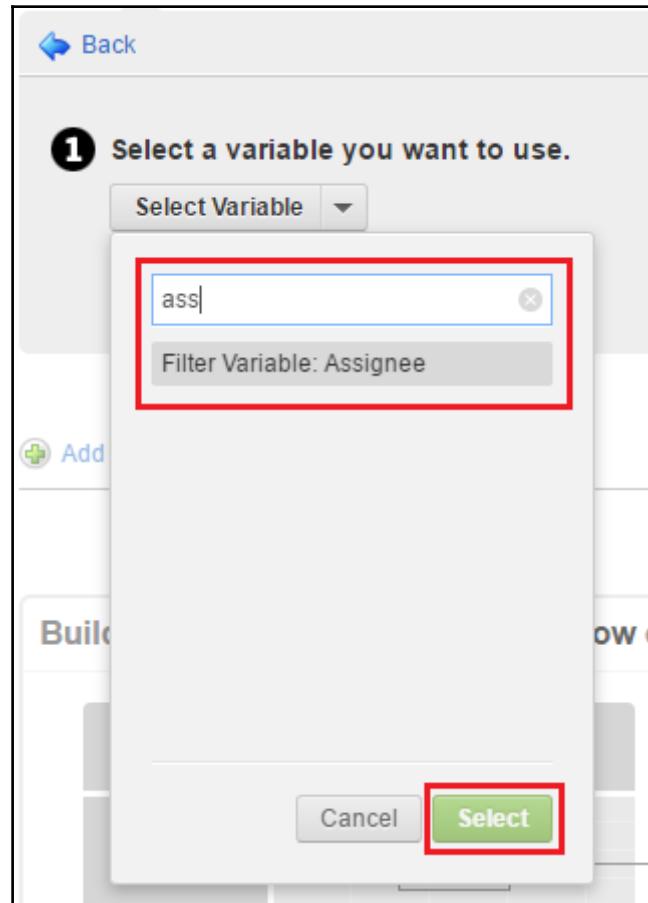
Ranking Filter
Rank your results. (Example: Top 10 Stores by Sales, Bottom 20 Outlets by Expenses).

Numeric Range Filter
Specify a range for your numbers. (Example: Stores where Sales is less than 10000 in 2008)

Variable Filter
Look for a variable. (Variables display dynamic data based on values stored in a specific user's profile.)

 [Add Filter](#) [Hide Filters](#)

Next, we will need to select the variable of our choice: **Filter Variable: Assignee:**



That is it. Whenever a user views the report, the variable filter will make sure that the report only displays the numbers according to the filter's attribute and the value assigned to it for this specific user.

Zendesk apps

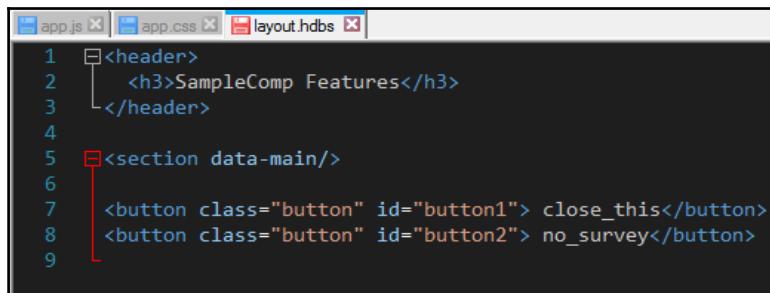
Whenever we miss a functionality in Zendesk, our minds should wander off towards creating a custom app. Let's review a few app ideas without going into too much detail.

Custom tagging app

Zendesk tags are very useful. However, adding and removing tags can be a tricky process, known to be prone to error. Too easily can we remove or misspell a tag without noticing.

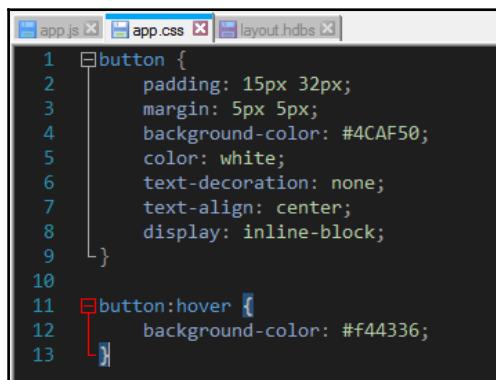
If adding and removing tags plays a part in your workflows, creating an app for it can make a huge difference.

We can use a simple HTML script to display two buttons on our layout:



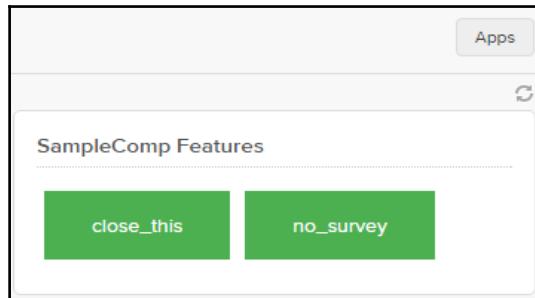
```
app.js app.css layout.hbds
1 <header>
2   <h3>SampleComp Features</h3>
3 </header>
4
5 <section data-main/>
6
7   <button class="button" id="button1"> close_this</button>
8   <button class="button" id="button2"> no_survey</button>
9 
```

Some CSS magic can add the final touches, making the buttons look less boring:



```
app.js app.css layout.hbds
1 button {
2   padding: 15px 32px;
3   margin: 5px 5px;
4   background-color: #4CAF50;
5   color: white;
6   text-decoration: none;
7   text-align: center;
8   display: inline-block;
9 }
10
11 button:hover {
12   background-color: #f44336;
13 }
```

The end product is not very impressive. I will leave the final design touches to you:



Let's move on to the JavaScript part of the app. The structure would look something like this:

```
app.js app.css layout.hbds
1 (function() {
2
3     return {
4         events: {
5             'app.activated':'doSomething',
6             'click #button1':'ADDclose_this',
7             'click #button2':'ADDno_feedback'
8         },
9
10    doSomething: function() {
11
12        //This is where we start
13        console.log("App started...");
14    },
15
16    ADDclose_this: function() {
17
18        //Add tag
19        console.log("Button1 pressed...");
20    },
21
22    ADDno_feedback: function() {
23
24        //Add tag
25        console.log("Button2 pressed...");
26    },
27
28    };
29 }());
```

A screenshot of a code editor showing the contents of the "app.js" file. The code is written in JavaScript and defines a function that returns an object containing event handlers and a "doSomething" method. The "doSomething" method logs "App started..." to the console. The "ADDclose_this" and "ADDno_feedback" methods log "Button1 pressed..." and "Button2 pressed..." respectively to the console. The code uses a syntax highlighting theme with red for comments and blue for functions and variables.

We start by listing our events and assigning a function to each one of them. The functions themselves are only placeholders outputting some text. After displaying our app and clicking on both buttons, we should see the following in our browser's console:

```
App started...
Button1 pressed...
Button2 pressed...
```

So what do we do next?

Next, we will have to add the actual functionality of adding tags.

But what if an agent clicks on the button by accident and needs to remove it again?

Good point! So the app should check whether the click is meant to add or remove a tag. We can do that by checking if the tag already exists:

```
16  ADDclose_this: function() {
17
18
19      //Log
20      console.log("Button1 pressed...");
21
22      //if the tag already exists...
23      if (this.ticket().tags().indexOf("close_this") != -1)
24      {
25          //...we remove it again
26          this.ticket().tags().remove("close_this");
27      }
28      //...otherwise...
29      else
30      {
31          //...we will add the tag
32          this.ticket().tags().add("close_this");
33      }
34 },
```

Now we can implement the same solution for the other button. Just go ahead and try it.

Once we are done with the basic functionality we can still get more creative and add additional features.

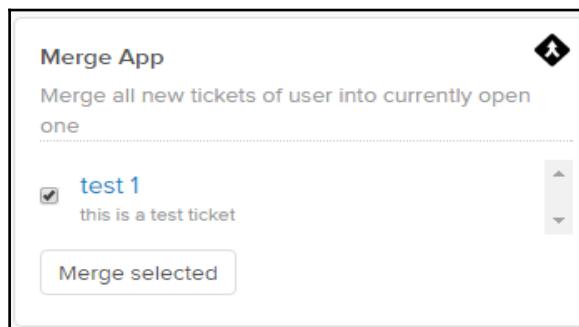
For instance, the button could change color once pressed.

Custom ticket merge app

Once in a while, end users send more than one ticket regarding the same issue. While Zendesk allows us to merge tickets, we will have to navigate away from the ticket to do so. We might also miss the duplicate tickets entirely.

How great would it be, if an app would not only inform us about the existence of more than one open ticket, but also allow us to merge them right on the spot?

The final version could look something like this:



The app would make use of two API calls:

- `queryNewTickets (" /api/v2/search.json" + search parameters)`
- `mergeTickets (" /api/v2/tickets/" + target ticket + "/merge.json")`

The first API call would take place as soon as we open the ticket. The app would search for any other open or new tickets by the requester.

If successful, the app would then commence by listing the tickets. We can now choose the tickets that we want to merge and do so by clicking on a button.

The second API call would then take care of merging the selected tickets into the open one.

Both API calls can be found in Zendesk's documentation for developers:

https://developer.zendesk.com/rest_api/docs/core/search

https://developer.zendesk.com/rest_api/docs/core/tickets

For those who are interested, this is how we could define the first API call:

```
15  requests: {
16    "queryNewTickets": function(user_email, status) {
17      return {
18        url: "/api/v2/search.json",
19        type: "GET",
20        dataType: "json",
21        data: "query=type:ticket " + status + " requester:" + user_email
22      };
23    },

```

We can use this API call later on in our `activated` function, which is called as soon as the app is being activated:

```
39  activated: function() {
40    return this.ajax("queryNewTickets", this.ticket().requester().email(),"status: new");
41  },

```

A big thank you to Franz Keferböck, who brought my first vision of this app to life and provided me with a great learning experience.

Custom macro statistics app

While Zendesk does provide us with a few statistics regarding our macro usage, creating a custom app can greatly improve our ability to create meaningful reports outside Zendesk.

So how do we achieve this?

We can go ahead and add an individual tag to each one of our macros. This however, could take a long time and the process, once again, is prone to error. So why not automate this process? Our app could go through all our macros and add the tag whenever it is missing. The actual tag could be a combination of the `macro_` keyword and the actual macro ID. This way, we can generate a direct link to the actual macro page by reading the id from the tag.

A finished version of the app could look something like this:

Admin Tools

[Set Macro IDs](#) [Create Macro Report](#)

LOG: Attempting to get macro list...

LOG: Getting more than one page...

LOG: Macro list(s) ready.

LOG: Done! Number of added tags: 0

LOG: Starting to search for tickets with tags.....

LOG: Report finished...

NAME	LINK	COUNT
Assign...	https://[REDACTED].zendesk.com/api/v2/macros/04903.json	10361
Temporary:: Premium	https://[REDACTED].zendesk.com/api/v2/macros/18325.json	12
Temporary:: leaderboard	https://[REDACTED].zendesk.com/api/v2/macros/17385.json	58
HARDWARE:: Smartwatches	https://[REDACTED].zendesk.com/api/v2/macros/82889.json	108
HARDWARE::Cannot import	https://[REDACTED].zendesk.com/api/v2/macros/46229.json	76
iOS::facebook connect issue	https://[REDACTED].zendesk.com/api/v2/macros/36949.json	65

Our app would consist of three main API calls:

- `get_all_macros (" /api/v2/macros/active.json")`
- `put_macro_tag (" /api/v2/macros/" + macroid + ".json")`
- `search_for_tickets_with_macro_tag (" /api/v2/search.json?query=tags" + tag)`

The first call would find all our active macros. We can then check whether a tag starting with `macro_` has been added already. If not, we can use the second call to add it accordingly.

The third call would allow us to find already tagged tickets in order to count how often each macro has been used. We can then move on to display a table, listing the macro's name, direct link for editing as well as the actual count. To make this feature more useful, we should think about adding a date-range filter.

How about some extra functionality?

We could, for instance, let our app check for macros that have not been used in a long time. It could list the macros and give us the option to deactivate them.

I would like to encourage you to come up with more helpful features. You can achieve almost anything utilizing the Zendesk's documentation for developers:

https://developer.zendesk.com/rest_api/docs/core/introduction

Custom Zendesk backup app

The idea behind this app is simple. You already learned about the different API calls available to us. Why not use them to back up our system?

The following could be requested via API and made available for download:

- Users
- Organizations
- Groups
- Views
- Business Rules
- Dynamic Content
- Macros
- Fields
- Targets
- Account Settings

We could even add some functionality to import them back into our Zendesk environment.

Zendesk CSS hack in apps

Zendesk's simplistic design is great, and in fact, there are not many reasons to tinker with the layout in the first place. But what if we want to change our ticket view, for instance?

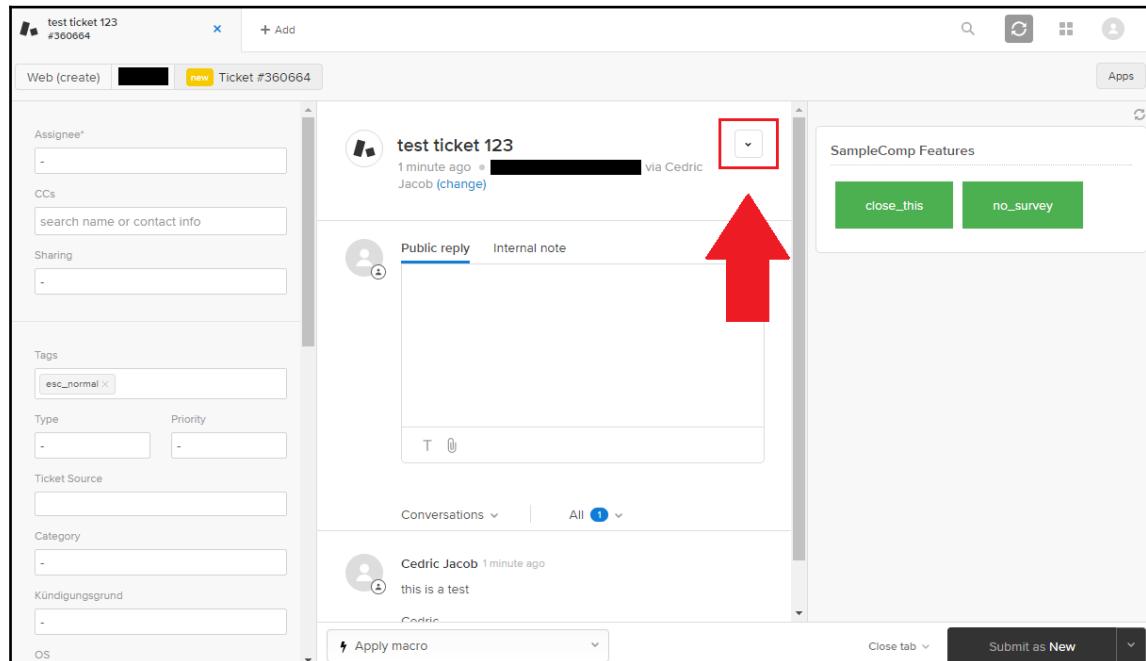
CSS is great for changing the pre-existing design and making it our own. There is only one problem. We might for instance want to try the following:

```
this.$('#element_id').css({  
  'border-width' : '5',  
  'border-color' : '#ff0000'  
});
```

This little code snippet should change the CCS rules for the element in question. But when trying to use this code in our Zendesk app to change an element outside of our app, it simply won't work.

This is because we are not allowed to access any element outside our given scope. Let's look at a more practical example.

This is the common ticket view running our ExampleComp Features app:



What if we wanted to hide the button marked in the preceding screenshot? Let's try the conventional method as discussed before:

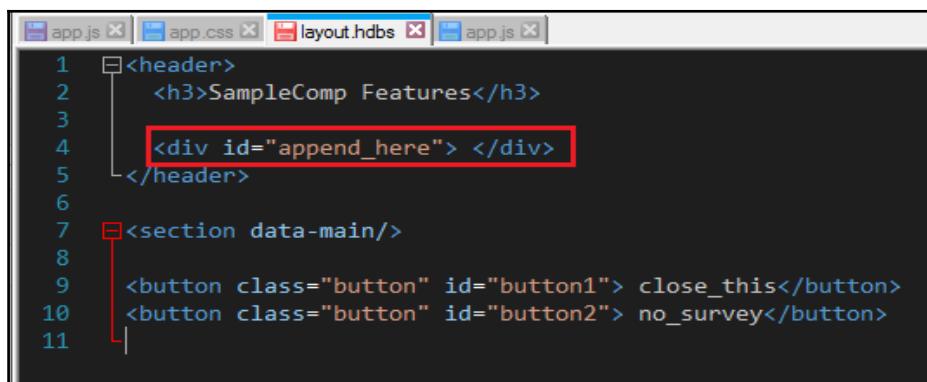
```
10  doSomething: function() {
11
12    //This is where we start
13    console.log("App started...");
14
15    //Hide Button (not working)
16    this$('.object_options_btn').css({
17      'display' : 'none'
18    });
19
20  },
```

While the code seems fine, the button still shows.

So what do we need to do in order for this to work?

The answer is rather simple. Instead of trying to access the element directly, we simply create the CSS rule as a string and append it to an HTML element that we are allowed to access.

Let's start by creating a new `div` in our layout:



A screenshot of a code editor showing a file named `layout.hbds`. The code contains the following HTML structure:

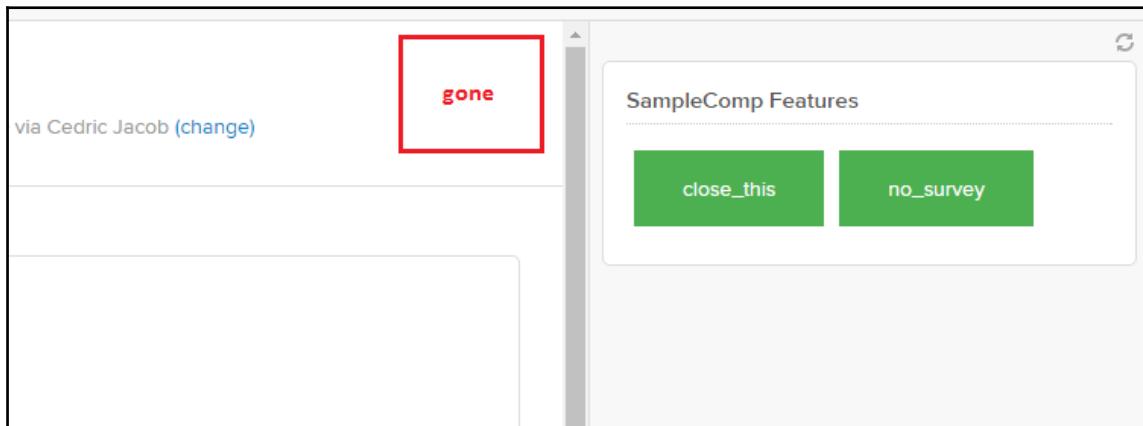
```
1 <header>
2   <h3>SampleComp Features</h3>
3
4   <div id="append_here"> </div>
5 </header>
6
7 <section data-main/>
8
9   <button class="button" id="button1"> close_this</button>
10  <button class="button" id="button2"> no_survey</button>
11
```

A red box highlights the `<div id="append_here">` element.

Next, we will need to change our JavaScript code:

```
10  doSomething: function() {
11
12      //This is where we start
13      console.log("App started...");
14
15      //Hide Button (not working)
16      this$('.object_options_btn').css({
17          'display' : 'none'
18      });
19
20      //Hide Button (working)
21      var _new_CSS_rules = '<style type="text/css">.object_options_btn { display: none !important; }</style>';
22      this$('#append_here').append( _new_CSS_rules );
23  },
```

When refreshing our app, the button should now disappear as expected:



Summary

In this chapter, we looked at some of my favorite tips and tricks when it comes to customizing Zendesk. We revisited some of Zendesk's most prominent components, learned how to utilize them in a different way, and consequently how to add more functionality to our environment. We dove a little deeper into the possibilities when it comes to custom apps, and I am sure some of you already have your own ideas and plans in mind.

I hope that by now you have a greater understanding of Zendesk's capabilities. I am glad of having had the opportunity to become a part of your Zendesk journey. Never forget to have fun with your setup and to think outside the box. There is so much more customizing to do.

Index

A

actions

- custom triggers, creating 161, 162, 163, 166
- Email group, notification 161
- Email user, notification 160
- requester- and organization-related actions 159
- ticket-related actions 154
- triggers, naming 166
- triggers, organizing 166
- triggers, testing 167

admin role 34

Advanced Security Enterprise Add-on

- about 331
- data, encryption 331
- enhanced disaster recovery 332
- HIPAA compliance 332

advisor role 34

agent-specific reports

- sending, variables used 367

agent

- about 34
- custom agent roles 34, 35, 36

Application Programming Interface (API) channel

- about 84, 85
- reference link 85

assignee 60

automations

- about 168, 169, 170
- custom automation, creating 177
- default automations 171
- specific conditions 175
- versus, triggers 120

B

breach 332

business rules

- about 119, 120, 347
- automations 339
- comments, adding to tickets via URL target 353
- SLAs 343
- tags, removing from tickets 350
- tags, used for disabling tickets 348
- tags, used for enabling tickets 348
- team leader, notifying of ticket with numerous replies 347
- ticket events, logging e-mail target used 357
- triggers 339
- troubleshooting 339
- used, for detecting user-errors 352
- used, for fixing user-errors 352

C

channels

- about 78
- API channel 84, 85
- Email channel 78, 79
- Facebook channel 80
- Mobile SDK channel 84
- Talk channel 82
- Twitter channel 80, 81
- Web Widget channel 83
- Zopim Chat channel 81, 82

Comma Separated Values (CSV) 56

conditions

- about 141
- current user 154
- organization custom fields 154
- requester-related conditions 152
- ticket-related conditions 141

contact support, ExampleComp

- Email 18
- Facebook 18
- Help Center 18

support form 18
Twitter 18
widget 18
Credit Card numbers
 redacting 329
 removing 329
CSS (Cascading Style Sheets) 210
custom agent roles
 about 34, 35, 36
 channels permission, granting 42
 example 43, 44
 Help Center permissions 40
 permission, granting to users 39, 40
 system permission, granting 43
 ticket permission, granting 36, 37, 38
 tool permission, granting 40, 41, 42
custom fields
 creating 67, 68
 custom organization fields 75
 custom ticket fields 71, 72, 73, 74, 75
 custom user fields 68, 69, 70
custom macro statistics app 376
custom metrics
 creating 295
custom organization fields 75
custom reports
 creating 284
custom roles
 creating, for staff members 361
 used, for reducing access 328
custom tagging app 372
custom ticket fields 71, 72, 73, 74, 75
custom ticket merge app 375
custom ticket views 181
custom user fields 68, 69, 70
custom Zendesk apps
 creating 203
 custom app, planning 203, 204
Ruby, installing 204, 205
Ruby, installing in Mac 206
Ruby, installing in Windows 206
ZAT gem, installing 206
Zendesk App Tools (ZAT), installing 204
custom Zendesk backup app 378

D

dashboard
 creating 270
 E-mails, scheduling 302
 filters, adding 276
 lines, adding 283
 reports, adding 272
 text, adding 283
 widgets, creating 280
default automations
 request customer satisfaction, rating 172
 ticket status set to solved, closing 171
default triggers
 about 123
 agents, notifying of received request 139
 assignee, notifying of assignment 133, 134
 assignee, notifying of comment update 130, 132
 assignee, notifying of reopened ticket 135, 136
 conclusion 140, 141
 group, notifying of assignment 136, 139
 requester, notifying of comment update 128, 129
 requester, notifying of received request 124, 126, 127, 128
Domain Keys Identified Mail (DKIM) 97
Domain Name System (DNS) 92

E

Email channel
 about 78, 79
 Accept wildcard emails 94
 custom domain for DKIM 96
Email templates 95
example, forwarding with Outlook.com 89, 90, 91
Gmail Go-to Actions 94
internal e-mail addresses, adding 86, 88
mail delimiter 96
Personalized email replies 94
settings 94
setup 86
SPF record, creating 92, 93
end-user 33
ExampleComp
 contact support 17

ticket's journey, monitoring 18, 19
existing user databases
 importing 54, 56, 57
 URL, for importing 56

F

Facebook channel
 about 80
 page, adding 97, 99
 setting 99, 100
 setup 97
field key 69
field title 69
fields
 about 58, 59
 Checkbox type 66
 Date field 66
 Decimal field 65
 Drop-down list 64
 Multi-line text 65
 Numeric field 65
 organization fields 62, 63
 Regular Expression field 66
 standard ticket fields 59
 Text field 64
 types 63, 64
 user fields 62, 63

G

general browser issues
 about 337
 browsing data, clearing 337
 cache, clearing 337
 extensions, disabling 337
 plugins, disabling 337
GoodData
 about 267, 362
 folders, report in 362
 reports, duplicating 364
 tags, report in 362
groups
 about 44, 45, 46, 47

H

Health Insurance Portability and Accountability Act (HIPPA)
 reference link 332
Help Center, Zendesk
 references 316
Help Centre channel
 about 111
 setup 111, 114
 support form 115, 116
history view
 creating 360
HTML (Hypertext Markup Language) 209

I

Insights
 about 267
 custom metrics, creating 295
 custom reports, creating 284
 dashboard E-mails, scheduling 302
 Dashboard, creating 270
 Tab, creating 270
installing, JIRA integration
 JIRA server, specifics 231
installing, Salesforce integration
 features, adding 243
 Salesforce data, viewing in Zendesk 237, 238, 239, 240, 241, 242, 243

J

JIRA integration
 about 221
 installing 228, 229, 230, 231
 preparing 222, 223, 224, 225, 226, 227
JSON Web Token (JWT) 314, 316
JWT(JSON Web Token) 305

M

macros 13
Mobile SDK channel 84
Multi-Dimension Analytical Query Language (MAQL) 295

N

Net Promoter Score
reference link 263

O

organization fields 62, 63
organizations
 about 47, 49, 50
 details 51
 domains 50
 example 51, 52
 group 50
 notes 51
 tags 50
 users 51

P

performance issues
 general browser issues 337
 Traceroute 338
 troubleshooting 334
 Zendesk status 335
private attachments 330

R

RailsInstaller
 URL, for downloading 206
reporting overview
 Community tab 260
 Insights tab 266
 Knowledge Base 258
 Leaderboard 257
 Net Promoter Score tab 263
 of Zendesk 252
 overview 253
 Satisfaction tab 264
 Search tab 261
reports
 about 362
 agent-specific reports, sending variables used
 367
 creating, in Zendesk 246
 duplicating, in GoodData 364
 exporting, in Zendesk 246

folders, report in GoodData 362
in Zendesk 246
tags, report in GoodData 362
tickets, excluding 364
requester 60
requester- and organization-related actions
 custom fields, organization 159
 custom fields, requester 159
 language, requester 159
requester-related conditions
 custom fields 153
 language 152
 role 153
 time zone 153
requirement evaluation 16
 ExampleComp 16
planning phase 17
road map, creating 30
Zendesk setup 25
roles
 about 359
 custom roles, creating for staff members 361
Ruby
 about 204
URL, for installing 206

S

Salesforce integration
 about 231
 installing 232, 233, 234, 235, 236, 237
 requisites 232
Secure Assertion Markup Language (SAML) 305, 314
secure coding practices 329
security
 accessing 305
 Advanced Security Enterprise Add-on 331
 best practice 304
 Credit Card numbers, redacting 329
 Credit Card numbers, removing 329
 custom roles, used for reducing access 328
 IP restrictions 327
 private attachments 330
 references 305
 secure coding practices 329

settings 304
single sign-on options (SSO) 312
two-factor authentication 310
Zendesk account, auditing 328
Zendesk password, authentication 306, 307, 308, 310
Sender Policy Framework (SPF) 91
Service Level Agreements (SLA) 14, 182
single sign-on (SSO)
 about 26, 312
 Enterprise SSO 313
 Secure Assertion Markup Language (SAML) 322
 social media SSO 312
staff 34
standard ticket fields
 about 59
 assignee 60
 CCs 60
 description field 60
 priority field 61
 requester 60
 share field 60
 status field 61
 subject field 60
 tags field 62
 type field 61

T

tabs
 creating 270
 filters, adding 276
 lines, adding 283
 reports, adding 272
 text, adding 283
 widget, adding 280

tags
 removing, from tickets 350
 satisfaction survey, disabling 350
 ticket, closing 349, 350
 used, for disabling business rules for tickets 348
 used, for enabling business rules for tickets 348

Talk channel 82

team leader 34

ticket events
 logging, via e-mail target 357

ticket's journey
 agent workflow 22, 23, 24
 ticket views 20

ticket-related actions
 assignee 157
 current user (CC), adding 158
 custom fields 159
 group 157
 priority 155
 satisfaction 157
 status 155
 tags, adding 158
 tags, removing 158
 tags, setting 158
 type 156

ticket-related conditions
 agent replies 151
 assignee 144, 145
 channel 147
 comment 150
 comment text 150
 custom fields 152
 group 144
 organization 146
 priority 143
 received at 148
 reopens 151
 requester 145
 satisfaction 148
 status 142
 tags 146
 type 142
 update 148
 values 149

ticketing workflows
 automations 12
 groups 14
 macros 13
 organizations 14
 Service Level Agreement (SLA) 14
 triggers 12
 views 13

Tier 1 Agent 36

Traceroute 338

triggers, and automations

- ticket events 339
trigger order 342
triggers
 about 121, 123
 actions 154
 conditions 141
 default triggers 123
 versus, automations 120
Twitter channel
 about 80, 81
 account, adding 100, 101
 account-related, settings 102, 103
 general settings 103, 104
 search, managing 104, 105
 setting 101, 102
 setup 100
- U**
- URL target
 comments, adding to tickets 353
user fields 62, 63
user tags 52, 53, 54
users 33
- V**
- vacation view
 creating 359
views
 about 13, 359
 history view, creating 360
 vacation view, creating 359
- W**
- Web Widget channel 83
Widget channel
 about 106
 customization 106, 107, 109
 setup 106
 snippet code 110
- Z**
- Zendesk account
 auditing 328
Zendesk App Framework SDK (ZAF SDK) 201
- Zendesk App Tools (ZAT) 204
Zendesk apps
 about 186, 187, 372
 Analytics & Reporting apps 191
 Channels apps 191
 Chat & Collaboration apps 191
 coding 215, 216, 217
 CSS hack 378
 custom macro statistics app 376
 custom tagging app 372
 custom ticket merge app 375
 custom Zendesk apps, creating 203
 custom Zendesk backup app 378
 E-Commerce & CRM apps 190
 Email & Social Media apps 191
 examples 193
 files, creating 206, 207, 208, 209, 210, 211, 212, 213
 Iframe app 199, 200, 201, 202
 installing 218, 219, 220
 IT & Project Management apps 190
 Knowledge & Content apps 190
 locations 191, 192, 193
 Out of Office app 202, 203
 previewing 213, 214
 Productivity & Time-Tracking apps 190
 Show Related Tickets app 193, 195, 196, 197, 198
 Surveys & Feedback apps 191
 Telephony & SMS apps 190
 troubleshooting 343
 uploading 218, 219, 220
 URL 188
 using, from Marketpalce 187, 188, 190
 Zendesk Labs apps 191
- Zendesk channels
 Email channel, setup 86
 Facebook channel, setup 97
 Help Centre channel 111
 setting up 86
 Twitter channel, setup 100
 Widget channel, setup 106
- Zendesk environment
 about 8, 10
 Main area 10

Sidebar 9
Top bar 9
Zendesk status
 about 335
 reference link 335
Zendesk support
 connecting 345
 URL 345
Zendesk
 about 7
 apps 12
 branding 15
 channels 10, 11
 efficiency 12
 fields 58, 59
GoodData 267
Insights 267
integration 12
localization 15
performance reporting 15
queries 25, 27, 28
reference link 53, 221, 376
reporting overview 252
reports 246
reports, creating in 246
reports, exporting 246
self-service 11
setup 10, 25
tasks, listing 29, 30
ticketing workflows 12
Zopim Chat channel 81, 82