Omni

Administrator Manual Team 6

Table of Contents

Introduction

Purpose of this Document

The intended audience for this document is any end users of Omni who plan to use Omni to write their own chat website or other chatting tool. This document will outline the procedures for setting up and properly maintaining an Omni chat server.

References

Omni User Interface Design Document

Havens, Micah, et al. “User Interface Design Document.” 9 Nov. 2022

Omni System Design Document

Havens, Micah, et al. “System Design Document.” 29 Oct. 2022, https://github.com/Skid-Team-6/Omnidocs/blob/main/Omni%20System%20Design%20Document.pdf

Omni System Requirements Specification Document

Havens, Micah, et al. “System Requirements Specification.” 20 Oct. 2022, https://github.com/Skid-Team-6/Omnidocs/blob/main/Omni%20System%20Requirements%20Specification.pdf

Omni Project Proposal

Havens, Micah, et al. “Project Proposal.” 12 Oct. 2022, https://github.com/Skid-Team-6/Omnidocs/blob/main/Omni%20Proposal.pdf

AM Template

Umrawal, Abhishek K., et al “Administrator Manual (AM) Instructions & Template”

Software Engineering 9

Sommerville, Ian, et al “Software Engineering 9th Edition”, March 2010, https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/index.html

System Overview

Background

The system to be maintained is a web server with its own database as well as the Omni server and its database. The system administrator will install and configure the system, run the system, and perform routine maintenance on the system.

Hardware and Software Requirements

To install and run Omni, it is recommended that you have a 64 bit multi-core processor and at least 4GB of memory. Exact memory and disk space requirements depend on your application of Omni. For small servers, we recommend 100GB of disk space. For large chat servers supporting upwards of one hundred (100) people, we recommend at least 8GB of memory and at least 500GB of disk space, depending on how often you would like to wipe messages, and preferably this space should be on a dedicated data storage server (such as a cloud service).

Administrative Procedures

Installation

Omni is written in Node.js, so you must have Node.js installed (along with NPM) to install it as a package. To use the Omni server as the backend for your chat site, the preferred method is to use `npm install Skid-Team-6/Omni`, though, if you would like, you can clone the GitHub repository from https://github.com/Skid-Team-6/Omni. Then, inside your module, do:

```
const Omni = await require("omni").create()
```

Note the use of "await" - this is because you can provide a backing store where Omni will save its data, which requires activating an Orbit coordinator before Omni can be used. Orbit is the database coordination tool that Omni uses. To provide a backing store, make sure you have a database server running that supports JSON in some manner (in this example we will use AJAX). You should always provide a backing store when not testing Omni.

```
const Omni = await require("omni").create(
      json_api = {
            host = "hostname.goes.here",
            namespace = "ajax"
      }
);
```

Omni primarily uses an in-memory store for storing channels, users, messages, etc, but if you provide a JSON API it will back up and sync its in-memory store with the JSON API you provide.

Routine Tasks

Checking the console log for errors should be done daily. Every week the system should be shut down and rebooted for maintenance. During this time you should backup the system, shut it down, and then reboot the system running off the new system. This is done because errors that do not crash the system can be fixed over this week but will not apply to the currently running system, thus a reboot is necessary.

To avoid filling storage space, periodic wipes of the database should be performed. It is sufficient to only remove messages from the database, and to only remove them once they are past a certain age threshold (such as at least one month old). This can be done by routinely making an API call to the JSON service you chose during the Installation process. Omni will not do this by default, as database management is left up to the user.

Periodic Administration

It is important to back up the user and message data of the server at least once a week, preferably more often. This is because if a crash were to occur, important messages and user information would be lost. To keep the losses minimal, it is important to backup as much as is realistically possible. Always provide a backing JSON API to Omni in a production environment, and back up whatever database is providing that API. Unused user accounts can be logged and cleaned, the requirements for this are up to the individual administrator.

User Support

Bugs in Omni can be reported on the GitHub issue tracker, where we will evaluate and respond to the report. See Dealing with Error Messages and Failures for more information. If any further support is needed, such as in the normal usage of Omni, submit this on the issue tracker, and depending on the question, we may either respond directly or we may add new content to the project wiki that answers your question.

Troubleshooting

Dealing with Error Messages and Failures

For crashes it is important to view the error message if provided, note it and then reboot the server from the last back-up. Once rebooted, you can work on fixing the error. Once the error is fixed, backup the system, shut it down, and then reboot it with the newly fixed system. For an error message that does not crash the server, you can begin fixing it immediately.

In many cases, an error will crash the server. In these cases, the error details will be printed out to the console. If the error is the fault of a bug in Omni’s code, please report it on the GitHub issue tracker at https://github.com/Skid-Team-6/Omni. Include with the report all relevant details that are printed to the console.

Known Bugs and Limitations

It is trivially easy to change a server's ID, and therefore to impersonate peer servers. Because of this, Omni's pairing feature is not production-ready. If you intend to use Omni in a production environment, do not use the pairing feature; in essence, never make an API call to any Omni endpoint that deals with peer requests. Peer-to-peer communication also happens over HTTP (not HTTPS), so it is not protected with TLS encryption. Please refer to the Code Inspection Report Document for more information on known deficiencies in Omni.

Omnichat, Team 6's front-end implementation, is for **reference only**. It is only a demonstration of what Omni can do, and should **never** be used in production due to numerous security concerns. Users have the ability to grant themselves administrator privileges, as well as to impersonate other users. If you intend to use Omni in production, use Omnichat as a reference for how Omni API calls should be made, but create a custom front-end with secure account management. It is possible that Omnichat will be made secure in the future, but this is not guaranteed.

Appendix A - Customer and Contractor Agreement

The customer and Team including: Micah Havens, Scott Devere, C.J. Commodore, Adnaan Dasoo, and Josh Martin are agreeing to the implementation of Omni in accordance with the information listed above. The team and the customer are agreeing that everything listed above is acceptable and sufficient for the task that the customer needs. If future changes need to be made to this document all members of the team will meet with the customer to explain what needs to be changed and why, and upon agreement the changes will follow.

Dated Signatures:

- Micah Havens, 12/11/22, X____MH____
- Scott Devere, 12/11/22, X______SD______
- C.J. Commodore, 12/11/22, X_____CC_______
- Adnaan Dasoo, 12/11/22, X_____AD______
- Josh Martin, 12/11/22, X_____JM_______

Customer Area:

Customer Comments:	
Date:__________	Signature:X___________

Appendix B - Team Review Sign-off

All members, including Micah Havens, Scott Devere, C.J. Commodore, Adnaan Dasoo, and Josh Martin, have reviewed the system requirements specification document for our software, named “Omni”. Each team member has reviewed this document for accuracy and completeness in all parts, including text, diagrams, bullets, charts, and tables.

Dated Signatures:

- Micah Havens, 12/11/22, X____MH____
- Scott Devere, 12/11/22, X______SD______
- C.J. Commodore, 12/11/22, X_____CC_______
- Adnaan Dasoo, 12/11/22, X_____AD______
- Josh Martin, 12/11/22, X_____JM_______

Appendix C - Document Contributions

- Micah Havens
    - Worked on: All sections
    - Percentage estimate: 25%
- Scott Devere
    - Worked on: Appendix, Introduction, Dealing with messages and Errors, Routine Tasks, Periodic Administration, and System Overview
    - Percentage estimate: 25%
- C.J. Commodore
    - Worked on: Routine Tasks, Periodic Administration
    - Percentage estimate: 15%
- Adnaan Dasoo
    - Worked on: Introduction, Known Bugs and Limitations
    - Percentage estimate: 20%
- Josh Martin
    - Worked on: Periodic Administration, User Support
    - Percentage estimate: 15%