

---

# **Tsune Documentation**

*Release 0.1a*

**dummydivision**

June 28, 2014



---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Getting ready to develop	3
1.2	What does Tsune mean?	3
<b>2</b>	<b>Setting up a development environment</b>	<b>5</b>
2.1	Requirements	5
2.2	Setting up Vagrant	5
2.3	Setting up the remote python interpreter	5
2.4	Configure Django integration	6
<b>3</b>	<b>Deploying Tsune...</b>	<b>7</b>
3.1	...to Heroku	7
3.2	...on Ubuntu Server	7
<b>4</b>	<b>Tsune Requirements and Dependencies</b>	<b>9</b>
4.1	Python Packages	9
4.2	Django Apps	9
<b>5</b>	<b>User FAQ</b>	<b>11</b>
5.1	How do I add a deck?	11
5.2	How do I add cards to a deck?	11
<b>6</b>	<b>Indices and tables</b>	<b>13</b>



This project is a web application for managing and learning flash cards.

Contents:



---

# Introduction

---

The goal of this project is to create a platform for people (especially students) to learn, create and share flashcards. There will be achievements, badges, duels and random pop-culture references.

## 1.1 Getting ready to develop

Our development setup includes VirtualBox, Vagrant and PyCharm and we use Heroku for deployment. See *Setting up a development environment* for a detailed guide.

## 1.2 What does Tsune mean?

“Tsune is usually used in a more mundane context, such as brushing one’s teeth regularly. But in the martial arts it refers to a state of such complete assimilation and involvement that training becomes a natural and automatic part of one’s daily life. Before tsune is achieved, the student often must struggle to make the practice schedule fit into his everyday lifestyle; an endless number of factors will seem to hinder and obstruct one’s practice routine. Some people are never able to accommodate regular training, and thus never pass this stage.

For those who persevere, however, tsune gradually takes over. A state is reached in which it is no longer necessary to think about disciplining one’s self to practice. The training becomes as automatic and as accepted as getting up in the morning, getting dressed and having breakfast. Now the feeling of awkwardness, the feeling that something is not right, comes only when a practice session is missed. At this point the martial artist has achieved a comfortable level of oneness with his art, and has fully integrated it into his life and his personality. This is also the happy situation which yields the greatest personal rewards. “

—Wendell E. Wilson (2010) ([Source](#))



---

# Setting up a development environment

---

This guide will help you set up a working development environment for Tsune

## 2.1 Requirements

1. A working and configured Git ([download](#))
2. VirtualBox compatible with Vagrant ([download](#))
3. Vagrant ([download](#))
4. PyCharm Professional ([download](#))
5. VirtualBox installation folder in PATH environment variable

## 2.2 Setting up Vagrant

1. Open PyCharm
2. Select *Open from Version Control*
3. Select either *Github* or *Git* and input the repository information
4. When asked whether PyCharm should open the project for you, select *Yes*
5. Select *Tools > Vagrant > Up* and wait for Vagrant to finish. This will take a long time (~30min), so grab a coffee or get some sleep. When it's finished, you will see *Process finished with exit code 0*

## 2.3 Setting up the remote python interpreter

2. Select *File > Settings* and click on *Project Interpreter > Configure Interpreters*
3. Click the + button on the right and select *Remote...*
4. Click *Fill from vagrant config* and *Ok*
5. Click *Test Connection*. Answer the authenticity warning with *Yes* and click *Ok*

6. Click *Ok* to close the interpreter setup and wait for the process to finish. This may take a few moments
7. When asked whether you want to set the interpreter as Project Interpreter, select *Yes*
8. Click *Ok* to close the settings window

## 2.4 Configure Django integration

1. Select *File > Settings* and click on *Django Support*. Make sure that *Enable Django Support* is ticked
2. Click the ... button next to *Settings*. Select *tsune/settings.py*. Click *Ok*
3. Close the Settings window with *Ok*
4. Click *Run > Edit Configurations*
5. If *Django server* has an entry named *tsune*, select the entry and continue with step 7
6. Click the + button. Select *Django server*
7. **Configure everything exactly as follows:** **Name:** *tsune* **Host:** *0.0.0.0* **Port:** *8000* **Run** **Browser:**  
*http://127.0.0.1:8080*
8. Close the Run configurations dialog by clicking *Ok*

---

# Deploying Tsune...

---

## 3.1 ...to Heroku

To deploy Tsune on Heroku, just push the repository to it:

```
$ git push heroku master
```

After that, you should set the SECRET\_KEY environment variable. To generate a new key, run the following from a python prompt:

```
from django.utils.crypto import get_random_string

chars = 'abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*(_=+)'
print get_random_string(50, chars)
```

Now set the SECRET\_KEY environment variable for your app. Run the following in your repository root:

```
heroku config:set SECRET_KEY="paste the generated key here"
```

## 3.2 ...on Ubuntu Server

1. `$ sudo apt-get update`
2. `$ sudo apt-get install chef`
3. Do not input anything when prompted to select a chef-server. Just press Enter.
4. `wget https://dl.dropboxusercontent.com/s/pcnysdzaie6wr58/postgres.json`
5. `chef-solo -j postgres.json -r https://dl.dropboxusercontent.com/s/fh3dxy0tbjuoulm/dependencies.json`
6. `sudo apt-get -y install libpq-dev python-dev firefox xvfb graphviz git-core`
7. `“git clone https://github.com/DummyDivision/Tsune “`
8. `cd Tsune && python manage.py syncdb && python manage.py migrate && python manage.py runserver`



---

# Tsune Requirements and Dependencies

---

This document lists all requirements and dependencies necessary to deploy Tsune. At the time of writing, all of them get downloaded automatically when you deploy out of the development environment.

## 4.1 Python Packages

1. [Django](#) is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
2. [Gunicorn](#) is a Python WSGI HTTP Server for UNIX.

## 4.2 Django Apps

1. [Django Guardian](#) is an implementation of per object permissions as authorization backend.



---

# User FAQ

---

## 5.1 How do I add a deck?

Click on the “+”-symbol in the “Deine Decks” overview.

## 5.2 How do I add cards to a deck?

Click on “Karten anzeigen” in the “Deine Decks” overview. All cards in the selected deck will be displayed. At the end of the last row of cards is a “+” Symbol. Click it to add a card.



---

# Indices and tables

---

- *genindex*
- *modindex*
- *search*