

---

# MIT RailSim

**Mojtaba Yousefi, Jonah Stadtmauer**

**Apr 17, 2024**



## CONTENTS:

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Project Overview . . . . .	1
1.2	Features . . . . .	1
1.3	License . . . . .	1
<b>2</b>	<b>Setup Guide</b>	<b>3</b>
2.1	Prerequisites . . . . .	3
2.2	Installing Pipenv . . . . .	3
2.3	Setting Up the Project Environment . . . . .	3
2.4	Activating the Virtual Environment . . . . .	3
2.5	Troubleshooting Python Version Errors . . . . .	4
2.6	Conclusion . . . . .	4
<b>3</b>	<b>Getting Started</b>	<b>5</b>
3.1	Installation Guide . . . . .	5
3.2	Quick Start . . . . .	5
<b>4</b>	<b>Usage</b>	<b>7</b>
4.1	Basic Usage . . . . .	7
4.2	Advanced Usage . . . . .	7
<b>5</b>	<b>Configuration</b>	<b>9</b>
5.1	Configuration Options . . . . .	9
5.2	Default Configuration . . . . .	9
<b>6</b>	<b>Architecture</b>	<b>11</b>
6.1	Components Overview . . . . .	11
6.2	Data Flow . . . . .	11
<b>7</b>	<b>Development</b>	<b>13</b>
7.1	Development Environment Setup . . . . .	13
7.2	Build Instructions . . . . .	13
7.3	Testing . . . . .	13
<b>8</b>	<b>Contribution Guidelines</b>	<b>15</b>
8.1	How to Contribute . . . . .	15
8.2	Community Guidelines . . . . .	15
<b>9</b>	<b>API Documentation</b>	<b>17</b>
9.1	API Overview . . . . .	17
9.2	Endpoints/Functions . . . . .	17

<b>10</b>	<b>Faqs</b>	<b>19</b>
<b>11</b>	<b>Troubleshooting</b>	<b>21</b>
11.1	Common Issues . . . . .	21
11.2	Getting Help . . . . .	21
<b>12</b>	<b>Changelog</b>	<b>23</b>
12.1	Version History . . . . .	23
<b>13</b>	<b>Appendix</b>	<b>25</b>
13.1	Glossary . . . . .	25
13.2	Further Reading . . . . .	25
<b>14</b>	<b>Indices and tables</b>	<b>27</b>

## INTRODUCTION

### 1.1 Project Overview

Briefly describe what the project does and its purpose.

### 1.2 Features

Highlight key features and capabilities.

### 1.3 License

Mention the project's license.



## SETUP GUIDE

This section outlines the steps for setting up the project environment using Pipenv, a tool for managing package dependencies and virtual environments. It assumes that the user has a working installation of pip.

### 2.1 Prerequisites

- Ensure pip is installed by running `pip --version` in your terminal. If pip is not installed, follow the instructions on the [pip installation guide](#).

### 2.2 Installing Pipenv

Pipenv can be installed via pip with the following command:

```
pip install --user pipenv
```

This command installs Pipenv for the current user. The `--user` flag ensures that Pipenv is installed in the user's directory and does not require system-wide installation.

### 2.3 Setting Up the Project Environment

Navigate to the project's root directory, where the `Pipfile` and `Pipfile.lock` are located, and execute the following command to install dependencies and set up the virtual environment:

```
pipenv install
```

### 2.4 Activating the Virtual Environment

Activate the virtual environment created by Pipenv using:

```
pipenv shell
```

## 2.5 Troubleshooting Python Version Errors

If you encounter an error about an unavailable Python version, such as:

Error: the specified Python version (3.8) is not available on your system.

It is recommended to use `pyenv` to manage multiple Python versions. Installation instructions for `pyenv` can be found at the [pyenv GitHub repository](#).

After installing `pyenv`, you may need to restart your shell or terminal to ensure `pyenv` and `pipenv` are correctly added to your `PATH`.

When `pyenv` is installed, running `pipenv install` again will prompt Pipenv to use `pyenv` to install the missing Python version. Confirming this allows Pipenv to manage the required Python version for the project automatically.

## 2.6 Conclusion

This guide provided instructions for setting up your project environment with Pipenv, including installing dependencies and resolving Python version issues with `pyenv`. For additional information, consult the Pipenv documentation or community forums.



## GETTING STARTED

### 3.1 Installation Guide

Provide step-by-step instructions for installing the project, including prerequisites.

### 3.2 Quick Start

A quick guide to getting a simple example up and running.



## **4.1 Basic Usage**

Explain how to use the project with simple examples.

## **4.2 Advanced Usage**

Dive into more complex use cases and features.



## CONFIGURATION

### 5.1 Configuration Options

Document configuration options, environment variables, etc.

### 5.2 Default Configuration

Explain the default setup and how to customize it.



## ARCHITECTURE

### 6.1 Components Overview

Describe the main components of the project.

### 6.2 Data Flow

Explain how data flows through the system.





## DEVELOPMENT

### 7.1 Development Environment Setup

Guide on setting up the development environment.

### 7.2 Build Instructions

Explain how to build the project from source.

### 7.3 Testing

Describe how to run tests.



## CONTRIBUTION GUIDELINES

### 8.1 How to Contribute

Instructions for making contributions, including coding standards and the pull request process.

### 8.2 Community Guidelines

Code of conduct and how to get involved in the community.



## API DOCUMENTATION

### 9.1 API Overview

High-level overview of the API.

### 9.2 Endpoints/Functions

Detailed descriptions of API endpoints/functions, including parameters, request/response formats, and examples.



---

**CHAPTER  
TEN**

---

**FAQS**

Address common questions and issues.





## **TROUBLESHOOTING**

### **11.1 Common Issues**

List common issues and their solutions.

### **11.2 Getting Help**

Information on where to ask questions or report issues.



## CHANGELOG

### 12.1 Version History

List of changes for each version, including new features, bug fixes, and breaking changes.



## **13.1 Glossary**

Definitions of terms used in the documentation.

## **13.2 Further Reading**

Links to additional resources such as blog posts, tutorials, and papers.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`