READ Developer Manual

Contents

[Installation Guide 2](#_Toc177644341)

[Django Backend 2](#_Toc177644342)

[React Frontend 2](#_Toc177644343)

[PostgreSQL Database 3](#_Toc177644344)

[Models 3](#_Toc177644345)

[User 3](#_Toc177644346)

[Story 3](#_Toc177644347)

[ReadingSession 3](#_Toc177644348)

[Class 3](#_Toc177644349)

[Student 3](#_Toc177644350)

[Authentication 4](#_Toc177644351)

## Installation Guide

The READ system makes use of a REST API framework, and so is separated into three main parts.

### Django Backend

Ensure you have the latest version of Python installed before proceeding.

#### Python Libraries

The backend application has a considerable number of open-source dependencies. These are stored in a file called **requirements.txt** and are easily installed using the following terminal command:

pip install -r requirements.txt

#### Other Dependencies

The following further installations are required for the application to function:

* **FFmpeg**, which can be installed from: <https://www.ffmpeg.org/download.html#build-windows>
* **Espeak-ng**, found at: <https://github.com/espeak-ng/espeak-ng>
  + Note that following installation you may need to add the following to your user / system variables:

PHONEMIZER\_ESPEAK\_LIBRARY = path\to\libespeak-ng.dll [including libespeak-ng.dll]

* + You’ll likely need to add the path to the *espeak-ng.exe* file to your path variables too.

### React Frontend

#### React Libraries

All React dependencies are in the file *package-lock.json* in the *read-frontend* folder, and are easily installed using the following terminal command:

npm install

#### Other Dependencies

The following further installations are required for the application to function:

* **Node.js**, which can be found at: <https://nodejs.org/en>

### PostgreSQL Database

If you do not intend to modify the current database functionality, the developers of the backend application recommend creating a PostgreSQL database called *users*.

#### Installation

**PostgreSQL** can be installed from: <https://www.postgresql.org/download/>

#### Database Creation

* Log into the PostgreSQL server
* Create a database called *users*

#### Schema Migration

* Migrating the current database structure is simple, and can be achieved using the following command:

python mange.py migrate

The database will need to be populated in order to test the application. One may achieve this by registering an *Admin* account and adding stories, then adding *Reader* and *Teacher* accounts to read stories and view reader progress respectively.

## Models

The application makes use of a few important models to store and manipulate data:

### User

The user model stores user details, including user role and reading level.

### Story

The story model stores details of the applications main content- stories.

### ReadingSession

The reading session model stores data relating to user sessions reading stories, including total time reading, number of errors and story progress.

### Class

The class model stores classes created by teacher users.

### Student

The student model stores enrolments by users into classes.

## Authentication

The application has extensive token-based endpoint and route protection, meaning that you will not be able to test much of the application without either creating users and logging in when necessary, or modifying the authentication mechanisms. The frontend application is currently set up to allow the registration of administrator users, though this functionality should, of course, not be including in production versions of the application.