**User Manual:**

**The Digital Backpack**

short line

**Table of Contents**

Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 2

Installation . . . . . . . . . . . . . . . . . . . . . . . . . 3

Django Server & PWA . . . . . . . . . . . . . . . . . . . 3

Android Application . . . . . . . . . . . . . . . . . . . 4

Configuration & Daily Operation . . . . . . . . . . . . . . . .5

Google Developer Console . . . . . . . . . . . . . . . . .5

Google Classroom . . . . . . . . . . . . . . . . . . . . .6

Running the Server . . . . . . . . . . . . . . . . . . . .6

Maintenance . . . . . . . . . . . . . . . . . . . . . . . . . .7

Server Certificates . . . . . . . . . . . . . . . . . . . 7

Troubleshooting . . . . . . . . . . . . . . . . . . . . . . . .7

Authentication . . . . . . . . . . . . . . . . . . . . . .7

Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . 8

**Introduction**

We are pleased that you have chosen DigiPack for your business needs. DigiPack is offered in two different applications, mobile and web applications. It is a helpful system that allows a fluid transition between online and offline learning which has been custom-designed to meet your needs.

Some of the key features for the mobile application includes:

* The ability to detect network connections.
* The ability to upload and download requests while offline.
* Two UI interfaces: Young adults and kids.
* Uses Google Interfaces:
  + Google Drive
  + Google Classroom
  + Google Search

The purpose of this user manual is to help you, the client, successfully install, administer, and maintain the DigiPack product in your actual business context in the future. We hope that the project will aid future generation’s education. The document will talk about the installation process, configuration, maintenance, and troubleshooting.

**Installation**

Over time, if it is necessary, you may want to move the product to a new platform or reinstall the product. This section discusses the installation process for the Digital Backpack. There are two major components of installation for the product. This is divided into the Django server installation and the android application installation. The progressive web app is included in the Django server.

**Django Server & PWA**

The Django server was originally hosted on a DigitalOcean Droplet, however, it should be easily compatible with other cloud hosting services as well. As previously mentioned, the progressive web app is implemented within the Django server, thus the installation process incorporates both of these components. The server was developed with Django version 3.1.6 and Python version 3.8.5 Steps for installation are as follows:

1. *SSH into your desired web server* - Once this is done you can create your desired directories and drag the Django project files into this directory.
2. *Create a Python virtual environment -* Ensure you are in the directory with the Django project files. Create a python virtual environment by entering the command:

**$ python3 -m venv [virtualenv name]**

1. *Enter your virtual environment -* Enter the virtual environment you just created by entering the following command:

**$ source [virtualenv name]/bin/activate**

1. *Install requirements -* A requirements.txt file is included within the Django server files. Install the requirements in the virtual environment using the following command:

**$ pip3 install -r requirements.txt**

1. *Make migrations* - The student database is handled using Django models. Make migrations to initialize the database with the following command:

**$ python3 manage.py migrate**

1. *Install SSL Certificates -* To secure the server with HTTPS, certificates are installed using Certbot from Let’s Encrypt. Customized instructions for using Certbot can be found on the Certbot website: <https://certbot.eff.org/>
2. *Run the server -* At this point, the installation process for Django is complete. The final step is to run the server. This can be done by running the following command:

**$ python3 manage.py runsslserver 0.0.0.0:8000**

**Android Application**

The Android Application can be built into an APK and installed that way or installed directly from the project in Android studio. This process covers how to install the Android application using both of these flows.

**Installing from an Android APK**

1. *Building the APK:* Retrieve the project source files from GitHub and open the project in Android Studio. Then, use the menu bar at the top of the screen and navigate to Build > APK(s) > Build APK(s). For now, opt for the APK to be signed with a debug key.
2. *Move the APK to your Android Device:* There are many ways to transfer files to Android; one way is connecting the device to a computer via USB.
3. *Enable Apps from Unknown Sources:* Navigate to the developer settings menu. If the developer settings menu is unavailable, navigate to *About Phone* and tap *Build Number* until developer mode is enabled on your device. Once in the developer settings menu, enable apps from unknown sources.
4. *Install the APK:* Navigate to the APK file in your phone’s file manager and launch the APK to install.

**Installing Directly from Android Studio**

1. *Enable USB Debugging:* Navigate to the developer settings menu. If the developer settings menu is unavailable, navigate to *About Phone* and tap *Build Number* until developer mode is enabled on your device. Once in the developer settings menu, enable USB debugging.
2. *Launch the App from Android Studio:* Open the project in Android studio and connect the Android device to your computer via USB. Use the drop-down menu next to the play button towards to top of the screen to select the physical Android device as the target. Press play and the app will be installed on the phone.

**Configuration & Daily Operation**

**Google Developer Console**

In order for the DigiPack authentication flow to function, some steps must be taken to register the system with Google. These operations are performed on the Google developer console which can be found at <https://console.developers.google.com/>. The steps that must be performed to get the Digital Backpack working are as follows.

1. *Creating a Project:* Create a new project on the Google developer console. This is where all authentication information relevant to the Digital Backpack will be stored.
2. *Activating Needed APIs:* The Digital Backpack makes use of multiple different APIs that are not enabled for a given project by default. Navigate to the API library and enable the following APIs for the Digital Backpack.
   1. Google Drive
   2. Google Classroom
3. *Configure the Consent Screen:* Google requires applications using its services to configure a consent screen that informs users of the nature of the project, permission requested, and who the user can contact with questions. Go to the Oauth Consent Screen tab to complete this process.
4. *Register the Proxy Server:* On the credentials tab, create a new OAuth2 client ID for a web application. Register the domain name for the web client as an authorized redirect URI. For example, <https://digipackweb.com:8000> is the authorized redirect URI for the current iteration of the proxy server. Download the JSON file and name it webClientSecret.json. Put this file in the server directory of the proxy server. In the android studio, navigate to the strings file and replace the string with id “serverClientId” with the Client ID associated with the credentials you just created.
5. *Register the Android Application:* The android application must be registered in the Google developer console as well. Every instance of Android Studio that is used to work on the app must be registered for versions of the app sources from that Android Studio instance to function.

**Google Classroom**

Google Classroom classes, students, teachers, and assignments created through the Google Classroom website ( <https://www.classroom.google.com> ) at this time cannot be used with the Digital Backpack application, however, this is not due to shortcomings of the Digital Backpack but of the Google Classroom service. When a classroom or assignment is created through the Google Classroom website, there are no credentials associated with the class/assignment and the associated credentials cannot be changed once a class is created. The only way to associate a project's credentials ( see: Configuration & Daily Operation - Google Developer Console ) is to manually create classes and assignments through the Google Classroom API.

Included in the DigiLearn GitHub repository under /demos/Google Classroom/ is a file named ClassroomSetup.py, it contains an example of the basic flow for creating a classroom, assignments, assigning a teacher and students to the class, as well as how to submit assignments for a user using the Google Classroom API and the DigiLearn Google Interface. Provided below are links to the Google Classroom API documentation and resources that explain more in detail how the Google Classroom API is able to be used.

Google Classroom REST API Documentation

<https://developers.google.com/classroom/reference/rest>

While the Digital Backpack uses the Python API ( linked below ), this page is the main reference for static values and JSON objects that all Google Classroom API’s return.

Google Classroom Python API Documentation

<https://googleapis.github.io/google-api-python-client/docs/dyn/classroom_v1.html>

**Running the Server**

After the initial setup of the server, running the server is as easy as entering a few commands into the SSH session terminal. The steps for running the server are as follows:

1. Change directories into the folder with the server files.

**$ cd [path to server files]**

1. Start the virtual environment

**$ source [name of virtualenv]/bin/activate**

1. Run the start server command\*

**$ python3 manage.py runsslserver 0.0.0.0:8000**

\*Note that the certificate paths may need to be specified for running the SSL server using the flags: --certificate & --key

**Maintenance**

**Server Certificates**

One component of the Digital Backpack is a web server that must maintain valid certificates in order to ensure security on the server. Aside from security reasons, HTTPS is absolutely necessary for the functionality of the progressive web app. The service workers utilized in the progressive web app only operate over HTTPS connections. As mentioned previously, certificates are handled on the server using Certbot from LetsEncrypt. Certificates from LetsEncrypt last 90 days, and can also be configured for auto-renewal. More information can be found in the LetsEncrypt documentation: <https://letsencrypt.org/docs/>

**Troubleshooting**

**Authentication**

There are some error codes likely to be produced during the installation and operation of the Digital Backpack that are related to authentication. Some of these error codes are benign and can be safely ignored. Others will need to be resolved for the DigiPack to function properly. These error codes are described below.

1. The Android App
   1. *Error Code 4:* Error code 4 indicates that the silent, automatic sign-in process has failed. This is likely a result of the user not signing in recently. This can be resolved by using the explicit sign-in flow instead.
   2. *Error Code 8:* Error code 8 stands for “Internal Error” and indicates that Google has, for some reason, failed to authenticate the Android device. This issue is usually resolved by retrying. This issue may be caused by the incorrect registration of the Android App. In that case, see “Google Developer Console” under “Configuration & Daily Operation.”
   3. *Error Code 10:* Error code 10 stands for “Developer Error” and indicates that there is an issue with how the Android App is configured in the Google Developer Console. See “Google Developer Console” under “Configuration & Daily Operation.”
2. The Proxy Server
   1. *Expired ID Token -* ID tokens are used to verify user connections to the server and are refreshed every time the user opens the app. In some cases, these ID tokens may expire before they can be redeemed by the server. This is a known shortcoming of the DigiPack in its current iteration. To resolve, restart the DigiPack app.
   2. *Flow Exchange Error -* Flow exchange error indicates that there was some issue exchanging the Authentication code for a credentials object. This is most likely a non-issue as the student will already be registered with the DigiPack system.
   3. *Refresh Token Error -* Refresh tokens are used to keep the credentials stored on the server valid over a long period of time. If the refresh token is expired or missing, the server cannot act on the behalf of the user. Resolve by unregistering and re-registering the user with the DigiPack.

**Conclusion**

With best wishes from your “DigiPack” developers: Sebastian Kastrul, Caitlin Abuel, Israel Bermudes, and Kristine Mae Hermosado. We thank you for allowing us to help the DigiPack vision come to life. It has been a great experience for the team to develop the application and work with you. While we are moving on to professional careers, we would be happy to answer short questions in the coming months to help you get the product deployed and operating optimally in your organization.

If you would like to contact us, our email addresses are:

Sebastian Kastrul: [sebastian.kastrul@gmail.com](mailto:sebastian.kastrul@gmail.com)

Caitlin Abuel: [csa73@nau.edu](mailto:csa73@nau.edu)

Israel Bermudes: [ilb8@nau.edu](mailto:ilb8@nau.edu)

Kristine Hermosado: [kh2336@nau.edu](mailto:kh2336@nau.edu)