**TECTONICA TOOLS GUIDE FOR CODERS**

Tectonica Tools (TT) is developed using the php based framework Laravel. It is hosted in AWS. The domain is registered in GoDaddy. The app uses Laravel 4.1 version (as specified in in composer.json)

**SIGN IN -LOGIN**

[A01]Arrive via get URL:

**Route:** get ('/account/sign-in' **Controller:** AccountController

**View:** signin **Method:** getSignin

[A02]Once you attempt to sign in into the application

 *(csrf)* **Route:** post ('/account/sign-in' **Controller:** AccountController

**View/Process:** **Method:** postSignin
Upon sign in, email and password are validated, if validation fails you are redirected to

get('/account/sign-in'

if validation passes you are sent to

get('dashboard'

**SIGN IN - RESET YOUR PASSWORD**

[A03]Arrive via get URL:

**Route:** get('forgot-your-password' **Controller:** AccountController

**View:** forgotYourPassword **Method:** getForgotPassword

[A04]Processing the form to recover password via Post:

 *(csrf)*

**Route:** post('forgot-your-password' **Controller:** AccountController

**View:** forgotYourPassword **Method:** postForgotPassword

Upon sign in, email is verified, if validation fails a flash message explaining that TT did not find a valid email (according to email formatting conventions), if email verification passes, the application checks if email inputted exists in the database, if so, TT generates a code and an associated password. An email is sent to the user with a link to a randomly generated password (with a link to get('/account/recover/{code}') , and redirecting the user to get('/account/sign-in' showing a message that says that email with new password has been sent. If the email is not found on the database the user is redirected to get('forgot-your-password',

[A05] Coming from the link given in the reset password email:

**Route:** get('/account/recover/{code}' **Controller:** AccountController

**View/Process: Method:** getRecover

Locates the code assigned to user, makes sure code matches the incoming user and looks that temp password is not set to empty for that user. Resets users' password and code, redirects visitors to get('/account/sign-in', and lets visitor know their account has been recovered please sign in with new password. Otherwise explains on message that it could not recover account and redirects to get('/account/sign-in',

**SIGN UP - CREATE ACCOUNT**

[A06]When you are not logged in and you want to create an account:

**Route:** get('/account/create' **Controller:** AccountController

**View:** create **Method:** getCreate

[A07]Once you are ready to create the account in the create view:

*(csrf)*

**Route:** post('/account/create' **Controller:** AccountController

**View/Process:** **Method:** postCreate

TT validates username, email and password. If validation fails, it redirects to get('/account/create'. If validation passes, TT stores account data into variables and creates an activation code for the user and associates it in an array to their data. Then sends an email to activate the code (linking to get('/account/activate/{code}'), redirects the visitor to get('/account/sign-in' and displays a message letting them know their account has been created and an email has been sent with code activation link.

[A08]Coming from the link given on the activation email:

**Route:** get('/account/activate/{code}' **Controller:** AccountController

**View/Process:** **Method:** getActivate

Switches the state of the user to active and redirects to get('/account/sign-in' and displays message saying account has been activated and user can start using the account. Then redirects to get('/' With a message that the app could not activate the account.

 **SIGN UP/ LOGIN WITH FACEBOOK AND GOOGLE**

[A09]

**Route:** get('login/fb' **Controller:** AccountController

**View/Process: Method:** loginWithFc

Upon clicking on the Facebook login/sign up button user is connected with the facebook url, which will request user login for facebook or check if user is logged in.

[A10]

R**oute:** get('login/fb/callback' **Controller:** AccountController

**View/Process: Method:** loginWithFcCallBack

Upon authorization from the user, the app receives the user’s information, verifies if the user’s email address exists in tectonica tools and retrieves the user’s information forcing a login to the application and redirecting them to the dashboard, or creating an account for the user if the user doesn’t have one. In this case an email is sent with a password and the user is assigned the username off of their email address. User is redirected to the dashboard with a message letting them know that their account has been created.

[A11] **Route:** get('googleLogin' **Controller:** AccountController

**View/Process: Method:** loginWithGoogle

Upon clicking on the Google login/sign up button user is connected with the google url, which will request user login for google or check if user is logged in.Upon authorization from the user, the app receives the user’s information, verifies if the user’s email address exists in tectonica tools and retrieves the user’s information forcing a login to the application and redirecting them to the dashboard, or creating an account for the user if the user doesn’t have one. In this case an email is sent with a password and the user is assigned the username off of their email address. User is redirected to the dashboard with a message letting them know that their account has been created.

 **NATIONBUILDER ACCOUNTS**

[B01] **Route:** get('accounts' **Controller:** AccountsController

**View:** accounts **Method:** index

Select from the database account information for the user, displays that info in the accounts view.

[B02] **Route:** get('addaccount' **Controller:** AccountsController

**View:** addaccounts **Method:** showNBAccount

**Accessed through the form on addaccounts.blade.php view:**

[B03] (csrf)

**Route:** post('addnbaccount', **Controller:** AccountsController

**View/Process:** **Method:** addNbAccount

Validates input, if validation fails, redirects to B02, addaccounts. If validation passes, redirects to nationbuilder url for authorization.

**ADD NATIONBUILDER ACCOUNT - AUTHORIZE**

[B04] **Route:** get('/get\_nb\_token' **Controller:** AccountsController

**View/Process:** addaccounts **Method:** processNbCode

Processes the code sent via input and insert into databes information pertaining to the code, user id, and the nationbuilder slug. Then show the view addaccounts and the flash message "Your NB account has been linked"

**DELETE NATIONBUILDER ACCOUNT**

[B05] **Route:** get('deleteNbAccount/{id}' **Controller:** AccountsController

**View/Process: Method:** deleteNbAccount

Selects all the nationbuilder account info for the user from the database. If there is no info to show, then redirects to 'accounts', if there is information deletes the information and returns visitor to 'accounts'.

**MAILCHIMP IMPORTER - RUN THIS APP**

[C01]: Arrive via get url:

**Route:** get('mailchimp-importer' **Controller:** ProcessAppController

**View/Process:** **Method:** mailchimpImporter

Saves a variable with directive to connect to mailchimp webservices

[C02]: Upon clicking "Run this App button":

**Route:** get('mailchimp-importer-step1' **Controller:** ProcessAppController

**View/Process: Method:** mailchimpImporterStep1

TT checks if there is a NB account associated with the user ID. If there isn't it redirects to get('addaccount' and shows a message saying that first you need to set up a NB Account.

**AVAILABLE APPS - ACCEPT AND SYNC NOW**

[C03]: Upon clicking on the "Authenticate Now Button"

**Route:** post('mailchimp-importer-step1-process' **Controller:** ProcessAppController

**View/Process: Method:** mailchimpImporterStep1Process

TT validates that checkbox is marked. If it is not it redirects to get('mailchimp-importer-step1' and flashes a container of errors to the session. If it is, it redirects to appropriate URL in Mailchimp Application.

[C04]: Upon logging in into Mailchimp

**Route:** get('mailchimp-importer-step2' **Controller:** ProcessAppController

**View/Process:** mailChimpImporterStep2 **Method:** mailchimpImporterStep2

TT retrieves the code and selects all database information from the client's Mailchimp account. Transfers the data from Mailchimp and displays the view with information from variables defined herein, including total lists, list data, and nationbuilder account information.

[C05] Upon clicking "Accept and Sync Now"

**Route:** post('mailchimp-importer-step3' **Controller:** ProcessAppController

**View/Process:** mailchimpImporterStep3 **Method:** mailchimpImporterStep3

TT transfers list members information and imports the data (first name, last name, email) into TT. It displays the view above.

**MAILCHIMP IMPORTER - TERMS AND CONDITIONS**

**[C06] Route:** get('terms-of-service' **Controller:** HomeController **View:** terms **Method:** terms

**The following item is not part of the application and will be removed in the future:**

[C00] **Route:** get('mailchimp\_importer\_complete' **Controller:** ProcessAppController

**View/Process: Method:** mailchimp\_importer\_complete

**GOOGLE CONTACTS IMPORTER - RUN THIS APP**

[D01] **Route:** get('google-importer' **Controller:** ProcessAppController

**View/Process: Method:** googleImporterStep1

Checks if user has a NB account, if they don’t, redirects them to get('addaccount', If they do, it takes them to view: googleContactsImporter

**INSTALLING GOOGLE CONTACTS IMPORTER - ACCEPT AND SYNCH NOW**

[D02] **Route:** post('google-importer-step1-process' **Controller:** ProcessAppController

**View/Process:** **Method:** googleImporterStep1Process

TT validates that checkbox is marked. If it is not it redirects to get('google-importer' and flashes a container of errors to the session. If it is, it redirects to appropriate URL in Google Contacts

[D03] **Route:** get('google-importer-step2' **Controller:** ProcessAppController

**View/Process:**googleContactsImporterStep2 **Method:** googleImporterStep2

Retrieves information from Google, checking if there is info for the NB account user, displays view googleContactsImporterStep2 prompting to proceed with synchronization.

[D04] **Route:** post('google-importer-step3' **Controller:** ProcessAppController

**View/Process:**googleContactsImporterStep3 **Method:** googleImporterStep3

Retrieves information from Google contact and imports it into Nationbuilder.

**GOOGLE CONTACTS IMPORTER - TERMS AND CONDITIONS**

[D5] **Route:** get('terms-of-service' **Controller:** HomeController

**View:** terms **Method:** terms

 **The following items:**

**Route:** get('importFileShow' **Controller:** ProcessAppController

**View/Process: Method:** ImportFileShow

 **Route:** get('importFileProcess' **Controller:** ProcessAppController

**View/Process:** **Method:** ImportFileProcess

**and the addPeople and createlist methods are currently not used by the application and will be removed in the future**

**CONSTANT CONTACT**

[E01] **Route:** get('constant-contact-step1', **Controller:** ProcessAppController

**View/Process:** **Method:** constantContactStep1

Checks that user has a nationbuilder accout. If they don’t, TT redirects them to get('addaccount', if they do, TT returns the 'constantContactImporterStep1' view.

[E02] **Route:** post('constant-contact-step1-process', **Controller:** ProcessAppController

**View/Process: Method:** constantContactStep1Process

Checks that the checkbox is marked, if it isn’t it redirects back to constant-contact-step1. If the checkbox is marked it connects with Constant contact and requests authorization when clicking “Authorize Now”.

[E03] **Route:** get('constant-contact-step2', **Controller:** ProcessAppController
**View/Process:** **Method:** constantContactStep2
Receives code from the the Constant Contact API, retrieving contact list information for the user. Displays constantContactImporterStep2 view. Otherwise returns to constant-contact-step1 route with an error message.

[E04] **Route:** get('constant-contact-step3', **Controller:** ProcessAppController
**View/Process:** **Method:** constantContactStep3
Actually imports contact information into Nationbuilder. Showing the constantContactImporterStep3

view if successful.

**OTHER DASHBOARD PROCESSES**

[R01] **Route:** get('/', **Controller:** DashboardController

**View/Process:** dashboard **Method:** index

Select all the apps that are activated and display them on ‘dashboard’ view.

[R02] **Route:** get('dashboard' **Controller:** DashboardController

**View/Process:** dashboard **Method:** index

User enters the application, TT checks that user has a nationbuilder account added. If they don’t it directs user to the add account screen with the message “First you need to add a nationbuilder account”. If they do have an account added, it shows the get('dashboard' view.

[R03] **Route:** get('installedApps' **Controller:** DashboardController

**View/Process:** **Method:** applicationsInstalled

Selects TT user’s applications installed and displays them at the ‘applicationsInstalled

 ‘ view, or if user has no applications installed, shows the ‘dashboard ‘ view.

**EMAIL AND PASSWORD RESET, LOGOUT**

(csrf)

[R04] **Route:** post('email-reset' **Controller:**AccountController

**View/Process: Method:** postEmailReset

New email is saved and email is sent to rest email in [R05]

Once you attempt to reset your email

[R05] **Route:** get('/account/email-reset/{code}' **Controller:** AccountController

**View/Process:** **Method:** emailReset

Resets the email and the code for the user and lets the visitor know their account has been recovered.

[R06] **Route:** post('password-reset' **Controller:** AccountController

**View/Process:** profile **Method:** postPasswordReset
Resets the password

[R07] **Route:** get('logout'
**View/Process: function** (){

 Auth**::**logout();

 **return** Redirect**::**to('/');

**PROFILE**

[R08] **Route:** get('profile' **Controller:** ProfileController

**View/Process:** profile **Method:** index

Displays Profile