**Smart\_weather navigations**

see smart\_weather\_naviations.txt

<Provider store={store}>

**<Drawer />**

</Provider>

**/\*\*\*\*\*\* Drawer \*\*\*\*\*/**

const Drawer = createDrawerNavigator(

//k this where you define routes

{

Current\_Weather: {path: '/', screen: **HomeScreen** },

// change

Family: {path: '/family', screen: **FamilyStack}**,

// to

Family: {path: ‘/family’, FamilyStack}

Friends: {path: '/friends', screen: FriendsStack },

Vacations: {path: '/vacations', screen: VacationsStack},

Works: { path: '/works', screen: WorksStack}

},

{

initialRouteName: 'Current\_Weather',

activeTintColor: '#e91e63'

}

)

// add

const FamilyStack = createStackNavigator({

Family: { path: ‘/family’, screen: FamilyScreen1, }

NewMember: NewMemberScreen,

})

**/\*\* HomeScreen \*\*/**

const MyNavScreen = ({ navigation, banner}) => (

<ScrollView>

<Button onPress={() => navigation.openDrawer()} title="Open drawer" />

<SmartWeather />

</ScrollView>

)

const HomeScreen = ({navigation}) => (

<MyNavScreen banner={'Home Screen'} navigation={navigation} />

)

// I didn't able to see this, when I click on "Curren5\_weather in the drawer."

HomeScreen.navigationOptions = {

}

\_onPress = (children) => {

alert(children)

}

**/\*\*\*\*\* smart\_weather \*\*\*\*/**

<View>

<View style={styles.row}>

<Text style={textStyles.mainText}>Forecast for </Text>

<View style={styles.zipContainer}>

<TextInput style={[textStyles.mainText, styles.zipCode]}

onSubmitEditing={this.\_handleTextChange} underlineColorAndroid="transparent" />

</View>

</View>

<View style={styles.row}>

<LocationButton onGetCoords={this.\_getForecastForCoords} />

</View>

{content}

</View>

let content = null

if (this.state.forecast !== null) {

content = (<View style={styles.row}>

<Forecast main={this.state.forecast.main} temp={this.state.forecast.temp} /></View>)

}

**/\*\* FamilyStack \*\*/**

const FamilyScreen1 = ({navigation}) => (

**<FamilyScreen />**

)

FamilyScreen1.navigationOptions = { headerTitle: 'Home Weather'}

const FamilyStack = createStackNavigator({

Family: { **screen: FamilyScreen1},**

})

FamilyStack.navigationOptions = {

drawerLabel: 'Family',

drawerIcon: ({tintColor}) => (

<MaterialIcons name="mood" size={24} style={{color: tintColor}} />

)

}

**/\*\*\* FamilyScreen \*\*\*/**

<View>

{this.\_mkMembersView()}

/\* to eliminate these to NewMemberScreen \*/

<LabeledInput label="Name" clearOnSubmit={true} onEntry={this.\_handlerName} onChange={this.\_handlerName} />

<LabeledInput label="zip" clearOnSubmit={true} onEntry={this.\_handlerZip} onChange={this.\_handlerZip} />

<Button onPress={() => this.props.navigation.navigate(‘NewMember’} title="New Member" />

</View>

// add

<Button onPress={this.\_newMemberScreen} title="+" />

\_newMemberScreen = () => {

let headerOptions = {

headerStyle: { backgroundColor: "#FFFFFF"},

title: "New Member",

headerLeft: <Logo />

}

this.props.navigation.navigate("NewMemberScreen")

}

const navigator = StackNavigator({

NewMember: { screen: NewMemberScreen, navigationOptions: headerOptions}

})

}

/\*\* create NewMemberScreen \*\*/

at root direct, because it will be re-usage from other screen

class NewMemberScreen extends Component {

render() {

return (

<View>

<LabeledInput label="Name" clearOnSubmit={true} onEntry={this.\_handlerName} onChange={this.\_handlerName} />

<LabeledInput label="zip" clearOnSubmit={true} onEntry={this.\_handlerZip} onChange={this.\_handlerZip} />

<Button onPress={this.props.\_createMember} title="Create Member" />

</View>

)

}

}

K Need to create NewMemberScreen first