

UROUD NERDS

React Native Module – Lists



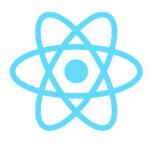
Peter Kassenaar – info@kassenaar.com



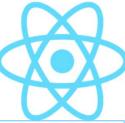
Lists in React Native

Lots of data – presented in a scrollable list

Looping over data



- Just like in regular React, using a .map() function
- Return a piece of UI per item
 - For instance, a <View><Text>...</Text></View>-block
- Don't forget to style the items
- We're using mock data here, from mockaroo.com
- Don't forget to wrap in a <ScrollView>, otherwise it will just flow of the screen



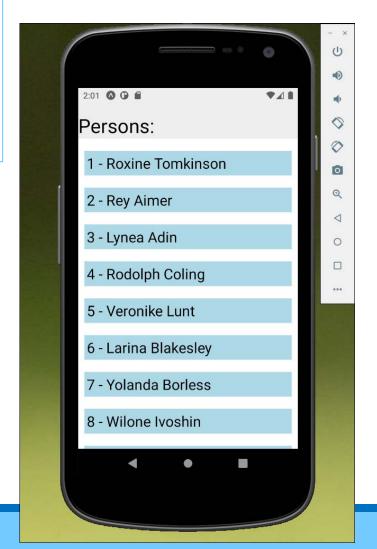
```
// Mock data, generated by https://www.mockaroo.com/
const persons_MOCK =
    [{"id": 1, "first_name": "Roxine", "last_name": "Tomkinson", "email": "rtomkinson0@foxnews.com"},
                                    import persons MOCK from "./assets/data/persons MOCK";
export default persons MOCK
                                    export default function App() {
                                        // O. state in React Native works like in React Web
                                        const [persons, setPersons] = useState(persons_MOCK)
                                        return (
                                            <View style={styles.container}>
                                                <Text style={styles.heading}>Persons:</Text>
                                                <ScrollView>
                                                        persons.map(person => (
                                                               <View key={person.id}>
                                                                   <Text style={styles.person}>
                                                                   {person.id}
                                                                 - {person.first name} {person.last name}</Text>
                                                               </View>
                                                </ScrollView>
                                            </View>
                                        );
```

Styling of list

```
const styles = StyleSheet.create({
          ...,
          person: {
                marginTop: 20,
                backgroundColor: 'lightblue',
                marginHorizontal: 10,
                fontSize: 24,
                padding: 4,
               }
        });
```



Styling of course is just a matter of personal preference, or UI-requirements/constraints





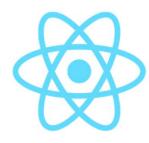
Using FlatList

Having more control over the generated list output

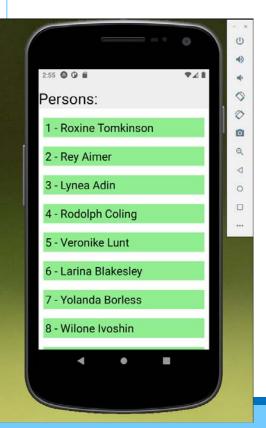
<FlatList />, More complex, more control

- Self-closing component, configured by props
- Mandatory props
 - data the data you want to output
 - renderItem function that renders each item
- renderItem receives an item property
 - which is every individual item in the array you loop over
 - No need for manual .map() function
- Results can look visually the same, but are more flexible

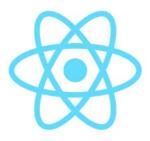
Using a <FlatList />



```
import persons MOCK from "../assets/data/persons MOCK";
const PersonFlatList = () => {
    const renderItem = (person) => {
        console.log(person)
        return (
            <Text style={styles.person}>
                {person.id} - {person.first_name} {person.last_name}
            </Text>
    return (
        <FlatList</pre>
            keyExtractor={(item) => item.id}
            data={persons MOCK}
            renderItem={({item}) => renderItem(item)}
        />
    1)
};
```



Properties



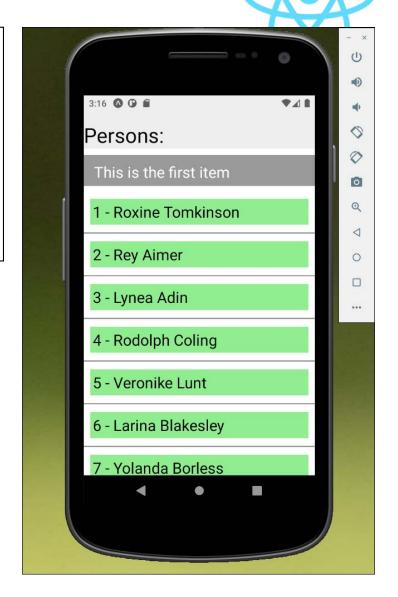
- keyExtractor tells <FlatList /> to use other property than key for it's key
 - here we return the id prop

Other props:

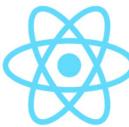
- ItemSeparatorComponent define component as separator
- numColumns how many columns?
- ListHeaderComponent, ListFooterComponent define components as beginning and end of the list respectively.
- Many, many more...
- https://reactnative.dev/docs/flatlist

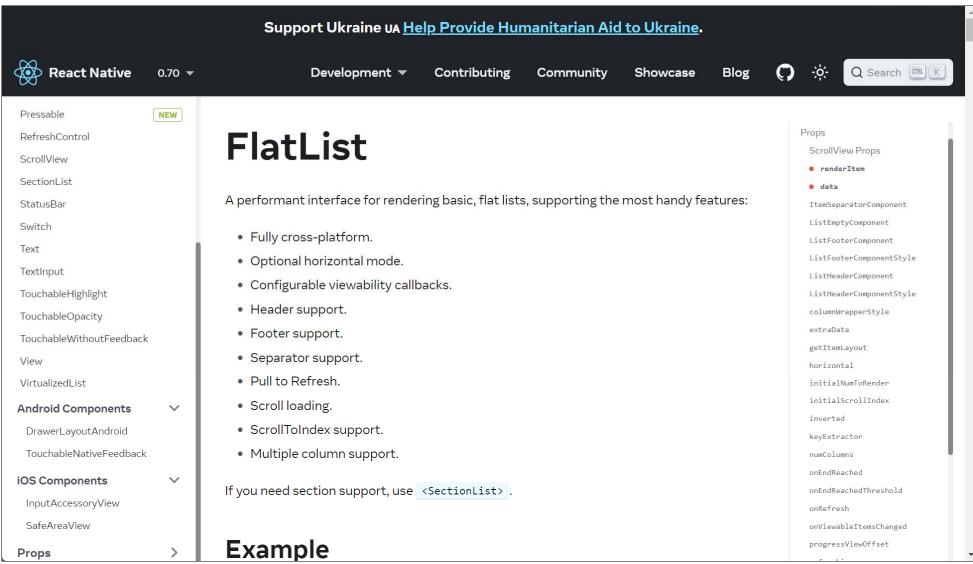
Example on more props for <FlatList />

```
<FlatList
   ListHeaderComponent={headerComponent}
   ListFooterComponent={footerComponent}
   keyExtractor={(item) => item.id}
   ItemSeparatorComponent={separator}
   data={persons_MOCK}
   renderItem={({item}) => renderItem(item)}
/>
```

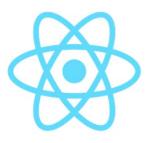


<FlatList /> API





Workshop



- Create your own mock data using Mockaroo (airport codes, cars, people, currencies etc)
- Show the data in a list on the screen using a <ScrollView>
- Update the component (or create a new one), so that a
 <FlatList /> is used.
- Experiment with the different properties for <FlatList />,
 - found at https://reactnative.dev/docs/flatlist
- Example code at ../50-scroll-flatlist.

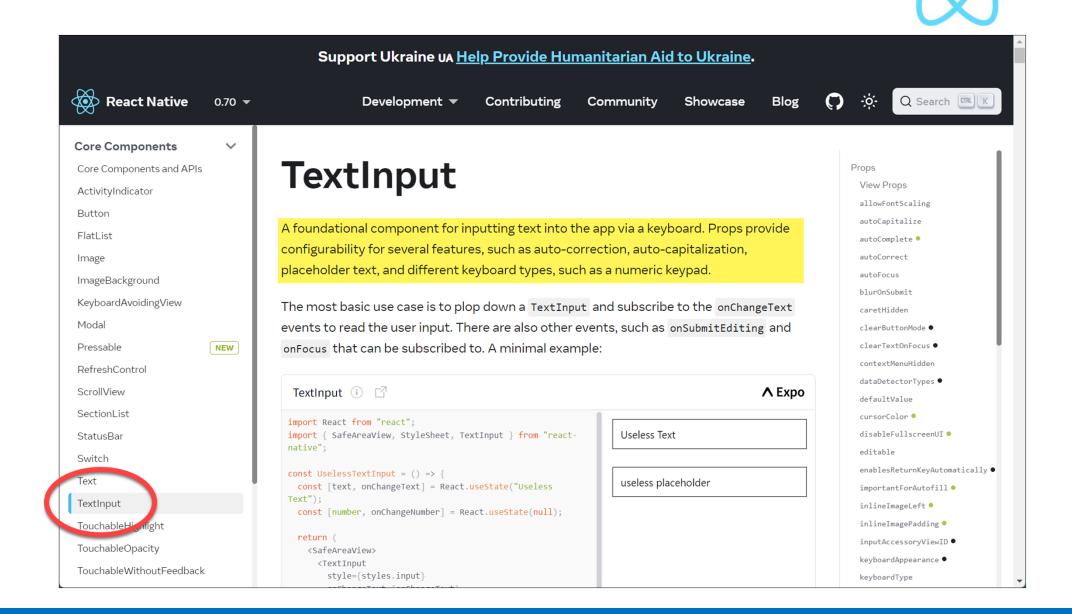
```
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
```



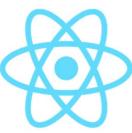
More info

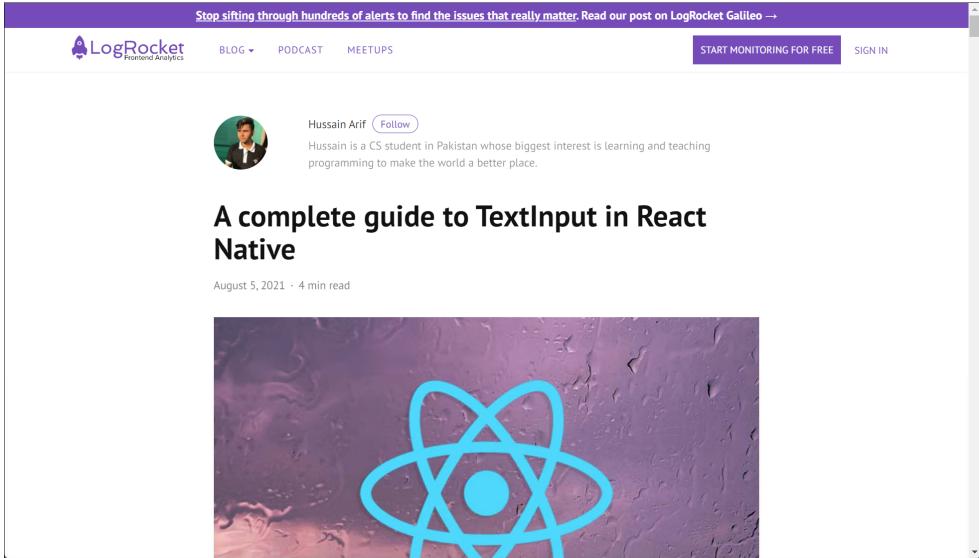
More info on the topics in this module

TextInput: https://reactnative.dev/docs/textinput

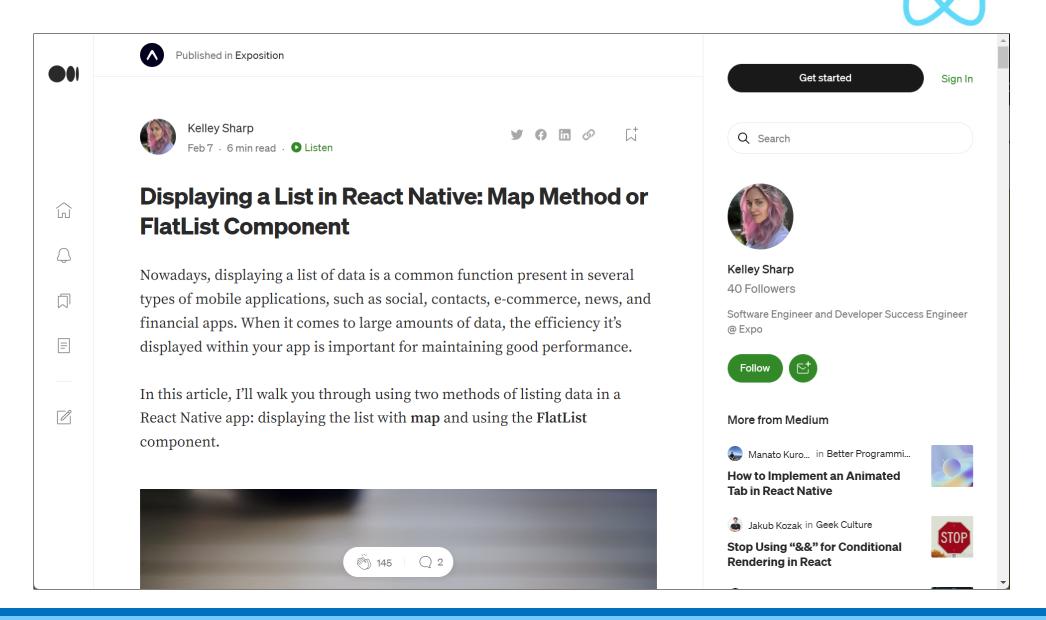


Blog on <TextInput />

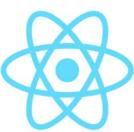


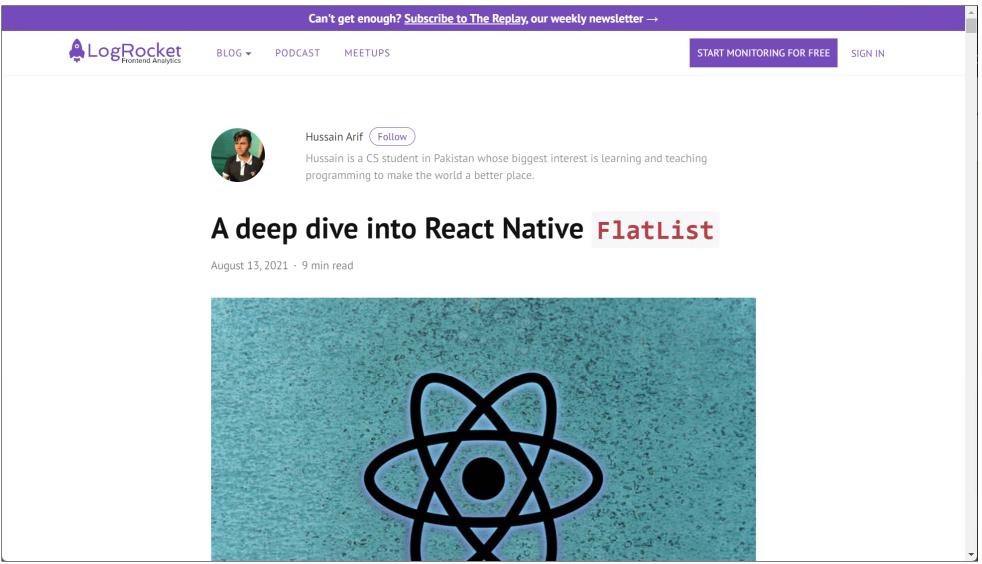


https://blog.expo.dev/react-native-flatlist-madeeasy-20fca51e0327



https://blog.logrocket.com/deep-dive-reactnative-flatlist/





Using the keyExtractor prop in FlatList

