Save, Display and Delete Your Pictures

Native App Development

Time: 60 mins

Introduction

In this class, the student/s will learn to save the captured pictures using an API call. They will also learn to fetch these images and display them as feeds on the feed screen. Student/s will also learn to delete any selected feed using the delete button and update the display on the feed screen.

New Commands Introduced

- method:POST   POST is used to send data to the to the API

- body: JSON.stringify({})   Pass data to the API

- headers{}   Set the data type of the data sent to the API

- useEffect()   useEffect is used to perform the side effects of a component. The actions to be performed before loading a component, after unmounting a component etc can be written within useEffect.

- async()   Species that the method has to be called asynchronously

- await   Specified before a fetch method makes a process to wait until the promise is returned

- method:DELETE   DELETE is used to delete data using and API call

Vocabulary

- Asynchronous function : Processes or functions can run in parallel. For example, an app may initiate an API call and continue with its processing without necessarily waiting for the API call to return.

Learning Objectives

Student/s should be able to:
- **_Recall_** the concept of API calls.
- **_Demonstrate_** how to save and later fetch the feed.
- **_Explain_** how to display the feeds fetched on the Feed screen.
- **_Add_** the functionality to delete a feed and update the Feed screen.

Activities

1. **Class Narrative:** *(2 mins)*
   - Brief the student/s that with the camera component added to the Snap Buzz app, we should be able to share the clicked pictures on the feed screen as feeds.

2. **Concept Introduction Activity:** *(5 mins)*
   - Let the student/s play the explore-activity and explore the Share feature of SnapBuzz app.
   - Ask students to click on Share and Delete buttons and observe how a caption can be added for each feed and then it can be deleted.
   - Using the slides, explain that the student/s will learn:
     - save the feed using an API call
     - fetch the feed using async await function call and render it on feed screen
     - delete a feed

3. **Activity 1:Share the Pictures:** *(12 mins)*

   Teacher Activity: *(6 mins)*
   - Explain how the pictures captured on camera in the app can be saved and accessed using a unique code unwritten in env.js.
   - Demonstrate how the images can be saved using the API function called addFeed(). Explain how method:POST specifies that data has to be saved using the fetch() method. Explain the wrapping of data in JSON format in the body attribute and content type in the headers attribute.

   Student Activity: *(6 mins)*
   - Guide the student/s to save a picture captured by them through the app camera.
   - Guide the students to change the screen name in App.js file so that the screen can be viewed in the preview pane and to match the opening and closing brackets in the function while coding.

4. **Activity 2: View Shared Pictures:** *(12 mins)*

   Teacher Activity: *(6 mins)*
   - Show how the pictures that were clicked are saved but the Feed screen still displays another set of pictures.
   - Explain how the images have to be fetched before the Feed Screen is rendered and the idea behind the need for success notification with a response of 200. Response has different status code: status code 200 means that the API call executed successfully whereas when 500 is returned it means that there was some internal server error. The code block for API call can be written within a try{}. A catch block would cater to any other errors that may occur during execution.
   - Explain in detail the function called useEffect() used to execute a set of statements before a component is rendered is React Native.
   - Explain that we have ensured that the screen render waits for the fetch to be completed before rendering which is achieved using asynchronous await function.
     Note: Explain async await using a real life example. Imagine that the tip of a pencil broke in the middle of writing something. The "process" of writing has to "wait" until the pencil is sharpened or another pencil is picked up. Similarly, await keyword makes sure the calling process waits until the async function returns a promise.

Student Activity: *(6 mins)*
- Guide the student/s to fetch the feed or images and display them on the feed screen.
- Guide the students to change the screen name in App.js file so that the screen can be viewed in the preview pane and to match the opening and closing brackets in the function while coding.

5. **Activity 3: Delete Shared Pictures:** *(14 mins)*

    Student Activity: *(14 mins)*
- Explain the need for the delete button to delete a specific feed from the feed upon clicking it on a given feed card.
- Explain to the students the use of the filter function in removing the selected feed card from the list of feed cards. Filter method is used to create a new array from an exciting array. The new array will have elements that satisfy the condition written in the filter function.

    currentFeeds.filter((feed) => feed.feedId !== feedItem.feedId)

    The new array currentFeeds will have the elements whose feedId is not equal to the feedId of the feed card to be deleted. Hence the card to be deleted will not be displayed on the Feed screen.

- Guide the student/s to delete the feed when the delete button is clicked using the API method called deleteFeed().

6. **Introduce the Post class project:** *(2 min)*
- Save and display the images taken with the camera in the Bid&Buy app.

7. **Test and Summarize the class learnings:** *(5 mins)*
- Check for understanding through quizzes and summarize learning after respective missions.
- Summarize the overall class learning towards the end of the class.

8. **Additional activities:**
- Encourage the student/s to change the color of the Like button when it is pressed.
- Encourage the student/s to save the status of Like permanently using an API call.

9. **State the Next Class Objective:** *(1 min)*
- We will learn to add a comment for each image or feed in.shared on the Snap Buzz App.

U.S. Standards:

CSTA: 2-AP-11, 2-AP-12, 2-AP-13, 2-AP-14, 2-AP-16, 2-AP-19

Links Table		
Activity	**Activity Name**	**Link**
Class Presentation	Save, Display and Delete Your Pictures	https://s3-whjr-curriculum-uploads.whjr.online/6c51f144-8772-453a-a22a-77dc8115bed4.html

Explore Activity	Save, Display and Delete Your Pictures	https://snack.expo.dev/@procodingclass/wad-402---54-sa3-delete-shared-pictures---solution
Teacher Activity 1	Save the Image	https://snack.expo.dev/@procodingclass/wad-402---54-ta1-save-the-image
Teacher Activity 1 Solution	Save the Image - Solution	https://snack.expo.dev/@procodingclass/wad-402---54-ta1-save-the-image---solution
Student Activity 1	Share the Pictures	https://snack.expo.dev/@procodingclass/wad-402---54-sa1-share-the-pictures
Teacher Reference: Student Activity 1 Solution	Share the Pictures - Solution	https://snack.expo.dev/@procodingclass/wad-402---54-sa1-share-the-pictures---solution
Teacher Activity 2	Get the Feeds	https://snack.expo.dev/@procodingclass/wad-402---54-ta2-get-the-feeds-
Teacher Activity 2 Solution	Get the Feeds - Solution	https://snack.expo.dev/@procodingclass/wad-402---54-ta2-get-the-feeds---solution
Student Activity 2	View Shared Pictures	https://snack.expo.dev/@procodingclass/wad-402---54-sa2-view-shared-pictures
Teacher Reference: Student Activity 2 Solution	View Shared Pictures - Solution	https://snack.expo.dev/@procodingclass/wad-402---54-sa2-view-shared-pictures---solution
Student Activity 3	Delete Shared Pictures	https://snack.expo.dev/@procodingclass/wad-402---54-sa3-delete-shared-pictures
Teacher Reference: Student Activity 3 Solution	Delete Shared Pictures - Solution	https://snack.expo.dev/@procodingclass/wad-402---54-sa3-delete-shared-pictures---solution
Student's Additional Activity 1	Toggle the Like Button	https://snack.expo.dev/@procodingclass/wad-402---54-aa1-toggle-the-like-button?platform=web
Teacher Reference: Student's Additional Activity 1 Solution	Toggle the Like Button - Solution	https://snack.expo.dev/@procodingclass/wad-402---54-aa1-toggle-the-like-button---solution?platform=web
Student's Additional Activity 2	Save the Status of Like	https://snack.expo.dev/@procodingclass/wad-402---54-aa2-save-the-status-of-like
Teacher Reference: Student's Additional Activity 2 Solution	Save the Status of Like - Solution	https://snack.expo.dev/@procodingclass/wad-402---54-aa2-save-the-status-of-like---solution
Post Class Project	Add and Show Auction Data	https://snack.expo.dev/@procodingclass/wad-402---54-pcp--add-and-show-auction-data?platform=web
Teacher Reference: Post Class	Add and Show Auction Data -	https://snack.expo.dev/@procoding

Project Solution	Solution	class/wad-402---54-pcp--add-and-show-auction-data---solution?platform=web