



Mobile Apps 3: Data, APIs and Permissions

Storage, APIs, JSON, status codes, offline and permissions

NAME _____

DATE _____

CLASS _____

Where App Data Lives

Local storage keeps data on the phone itself. It works with no internet and is fast, but it stays on that one device, and it is lost if the app is deleted. **Cloud storage** keeps data on a server on the internet. It can be shared with other users and synced across a person's devices, but the app needs a network connection to reach it.

Talking to a Server

Apps get cloud data through an **API** (Application Programming Interface): a set of rules for asking a server for data. The app sends a **request** to a URL called an **endpoint**. A **GET** request asks the server to send data back. The server replies with a **response**, which usually holds data in **JSON** (JavaScript Object Notation): "key": value pairs inside curly braces { }, and lists inside square brackets []. In code, list items are counted from 0, so events[0] is the first item in the events list.

Every response includes an **HTTP status code** that says how the request went:

200 OK

It worked. The data is in the response.

401 Unauthorized

The request needs a valid login, or the user's login has expired.

404 Not Found

Nothing exists at that URL, often because of a wrong address or a deleted item.

500 Server Error

The server had a problem. It is not the user's fault; trying again later may work.

Phones lose signal often, so good apps plan for being **offline**. They keep a **cache** (a saved local copy of the last data downloaded) and show it with a note that it may be out of date. They show a clear **error message** that says what happened and what to do, never a blank screen or an endless spinner.

1 Local or Cloud?

Write **Local** or **Cloud** for each kind of data, with a short reason based on the definitions.

Data	Local / Cloud	Reason
a. A chat message a friend must see on their phone		
b. A "dark mode on" setting for this phone only		
c. A to-do list the user wants on phone and tablet		



Mobile Apps 3: Data, APIs and Permissions (page 2)

2 Read the JSON Response

A school events app sends this request: `GET https://api.example.com/events?date=2026-11-05`. The server answers with status **200** and this JSON:

```
{
  "date": "2026-11-05",
  "count": 3,
  "events": [
    { "id": 101, "title": "Robotics Club",
      "room": "B12", "start": "15:30",
      "free": true },
    { "id": 102, "title": "Fall Play",
      "room": "Auditorium", "start": "18:00",
      "free": false, "price": 5 },
    { "id": 103, "title": "Math Tutoring",
      "room": "Library", "start": "15:15",
      "free": true }
  ]
}
```

a. How many events are in the response? Which key tells you?

b. What is the title of event `102`, and what does it cost?

c. Which event starts earliest? (Times use a 24-hour clock.)

e. Write the code path to the room of the **first** event in the list.

f. Only event 102 has a `price` key. What could go wrong if the app tries to display `price` for every event?

d. The app shows a "FREE" badge on some events. Which key should it check?

3 Status Codes

For each situation, write the status code the server would send (200, 401, 404 or 500) and a short, helpful message the app should show the user (or "none" if no message is needed).

Situation	Code	Message the app shows
a. The events list loads correctly		
b. The user's login expired last night		
c. The user taps an event that was deleted yesterday		
d. The school's server crashed while answering		



Mobile Apps 3: Data, APIs and Permissions (page 3)

4 No Signal

Ana opens the events app on a train with no signal. She used it on Wi-Fi yesterday. **a.** Circle the best behavior, then explain your choice using the word **cache**.

A. Blank screen **B.** Endless spinner **C.** Yesterday's events + note "Offline, saved yesterday 7:42 AM"

b. Ana taps **Buy Ticket**, which needs the server. Write the error message the app should show.

Permissions and Privacy

Phones protect sensitive data and hardware, like the **camera**, **microphone**, **location** and **contacts**. An app must get the user's OK through a **runtime permission**: a system pop-up shown while the app is running. Users can allow or deny it and change it later in settings. Both iOS and Android let users share **approximate location** (roughly where they are) instead of **precise location**.

Good apps follow the **principle of least privilege**: ask only for the access the app truly needs. They ask **at the moment of need** (when the user taps the feature that needs it), not all at once at launch, and explain the **reason** first. If the user says no, the rest of the app should still work.

5 Judge the Request

Is each request reasonable? Write **Yes** or **No**, then give a reason using the ideas above.

App and request	Yes / No	Reason
a. Flashlight app: contacts, at first launch		
b. QR-code scanner: camera, when user taps Scan		
c. Photo filter app: microphone		
d. Weather app: approximate location, on tap		

6 Design a Permission Request

Your study-spot app shows quiet places nearby, so it would like the user's location.

a. When will the app ask?

c. Reason message shown before the pop-up:

b. Precise or approximate? Why?

d. What happens if the user says no?