Sample iOS Project

by Devmountain Tutorials

WHAT'S COVERED

In this lesson, you will learn how to analyze the steps that the iOS engineer takes to make sure that the mobile app works effectively for the user. Specifically, this lesson will cover:

1. Adding Other Pets to the App

When Poodle Jumper first launched, we only provided services for dogs, but we are expanding to include services for cats, rabbits, hamsters, gerbils, and mice. Last week I finished the changes to the screens, or views as they are known in iOS. Today, I am working on pulling in the new data structure into the app so customers can add their additional pets. I'm going to show you some of that process.

STEP BY STEP

I start by looking at:

1. The existing data model I currently have in the app
2. The new data model from Monique, our Software Engineer
3. The updated web service from Ruben, our Web Developer
4. The requirements from Jose, our Product Manager
5. The designs from Mori, our UX Designer

I compare the changes and figure out how to structure them in the app. Do I need to pull data in a certain sequence? If so, how can I write that code efficiently to be as fast as possible and only include the data I need? When I have a plan for how to execute this in the app, I collaborate with Ruben (Web Dev) and Monique (Software Engineer) because they provide parts of the puzzle that I need. Monique creates the data model and the main database where we store information. Ruben creates a **web service** for the apps to get that data from the database.

TERM TO KNOW

Web service

A type of electronic communication between devices used to send and receive information, or data.

2. Structuring the Data

The web service Ruben created has all the pet fields (name, species, breed, size, type of food, attention needed, and special care notes) in the same service as the user's profile. This means that every time I need to refresh one piece of data for a pet, I have to get all the information for the user's profile and all the other pets. This isn't the most efficient way to structure the data, so I have a quick conversation with Ruben (Web Dev) and Camilla (iOS). We come up with a way to break the one big service into smaller microservices.

☆ BIG IDEA

Remember, the more data coming in and out of the app, the more networking and data usage it eats up while draining the device battery.

Thinking About Web Service

Knowing the structure of the data allows me to plan how and when I make the app send a request to get the data. This comes with its own set of challenges to overcome.

- What should the app do if the user is offline or doesn't have a stable internet connection?
- What data should I save, or cache, to improve the performance?
- If I cache the data, how often does it need to be refreshed?
- Should I refresh it when the view loads, or will users pull down on the screen to make it refresh?
- What do I show the user if the web service sends back an error message?
- Do I need to wait for the data to return before I show the view?
- What do I need to do to make sure the data is in sync with all their other devices?

All of these questions are considerations I have while writing the code. My goal is to create an experience that is as smooth as possible. I can't lock up the user's device while it loads, so I leverage the phone's multitasking abilities to have several things processing at the same time.

SUMMARY

This lesson explored the process of **adding new pets to an iOS app**, with a focus on **structuring data** efficiently. The data model was compared with the new data from a software engineer, web developer, product manager, and UX designer. The web service created by the web developer has all the pet fields and user profiles in one service, which isn't efficient. The iOS developer collaborated with the web developer to break the service into smaller microservices. The challenges of planning how and when to make the app send a request to get the data and how to handle errors and refresh the data are also discussed. The goal is to create a smooth experience that leverages the phone's multitasking abilities.

Source: This tutorial was authored by DEVMOUNTAIN and Sophia Learning. Please see our Terms of Use.

TERMS TO KNOW

Web service

A type of electronic communication between devices used to send and receive information, or data.