

Ajatempel	How long have you been an iOS software developer?	How long have you been an iOS software developer?	If there is an aspect of the iOS SDK that immediately makes you think of using an external third party framework, which ones are they?	Can you specify why?	Have you used any of these networking related frameworks?	Can you specify why?	Have you used any of these data storing frameworks?	Can you specify why?	Have you used any of these testing frameworks?	Can you specify why?	Have you used any of these view related frameworks?
02/12/14 10:18	3 to 5 years	3 to 5 years	Network, Graphics, Data Store, Social	Sometimes apple doesnt provide what you need for this "aspects"	ASIHTTPRequest, JSONKit, json-framework	To try the efficiency and simplicity to use					
02/12/14 10:30	6 or more years	3 to 5 years	Network, Data Store, Social, Utilities	Good external frameworks make several SDK features easier + faster to use. Less boilerplate code.	AFNetworking, RestKit, ASIHTTPRequest, JSONKit		FMDB, mogenerator	Mogenerator for adding custom methods to objects. FMDB for direct access to sqllite database.	GHUnit	For replacing default testing framework in xCode.	
02/12/14 10:35	1 to 2 years	1 to 2 years	Network, Social, Utilities	AFNetworking for handling networks is very robust and simple.	AFNetworking, ASIHTTPRequest	They provide simple interfaces to make requests.	mogenerator	Simplifies usual overhead that occurs when adding/removing/modifying database attributes and entities.			iCarousel
2/13/2014 11:54:18	3 to 5 years	1 to 2 years	Data Store	Third party frameworks can make using data store much more convenient.			MagicalRecord, mogenerator	More convenient use of data store			
2/18/2014 1:21:34	3 to 5 years	1 to 2 years	Network, Data Store, Social	For these I've found that whether these parts of APIs are impossible to use in order to solve some problem or you just have to write a lot more code to establish result. Social networks remote APIs change quite fast that it's hard for Apple to keep track and update libraries.	AFNetworking, ASIHTTPRequest	AFNetworking is handy for RESTful services with it's powerful serialisation interface. I do like heavy use of block based approach that makes me to avoid implementing tons of protocol methods. From ASIHTTPRequest most useful for me have been ASIDownloadCache. Caching to disk is made that simple.	FMDB	FMDB helps to reuse already existing SQL Queue helps to work around problems introduced by multithreading.	GHUnit	Allows unit testing also UI testing. For me have decent mocking support for network but still it can be improved.	
2/20/2014 14:56:28	6 or more years	3 to 5 years	Network, Views	Two things that I find myself often writing boiler-plate code for.	AFNetworking, JSONKit						FlatUIKit
2/21/2014 9:11:23	6 or more years	3 to 5 years	Network, Graphics, Data Store, Debug, Utilities	Have simpler usage, less code.	AFNetworking, ASIHTTPRequest, SDWebImage, JSONKit, SDWebImage	Have simpler usage, less code.	MagicalRecord, FMDB	No stupid C code to manage Sqlite database.	GHUnit		ViewDeck/JASidePanels/ECSlidingViewController
2/26/2014 13:33:33	6 or more years	3 to 5 years	Network, Social, Debug, Utilities	Third party components provide a higher level abstraction on top of what Apple is providing.	AFNetworking, RestKit, ASIHTTPRequest	These have provided a simpler interface for networking. Some of these (json-framework) were used, but have gained a better built in solution.	mogenerator		Kiwi, GHUnit		FastImageCache
2/27/2014 15:49:48	3 to 5 years	3 to 5 years	Network, Data Store, Utilities		AFNetworking, ASIHTTPRequest, SDWebImage, JSONKit, SDWebImage		MagicalRecord, FMDB				ViewDeck/JASidePanels/ECSlidingViewController, PaperFold-for-iOS, DTCoreText
2/27/2014 16:06:26	6 or more years	3 to 5 years			AFNetworking, RestKit, ASIHTTPRequest, JSONKit	Early on the iOS SDK did not have as robust networking capabilities as it has now so things like RestKit and ASIHTTP were more useful, but as the SDK has grown and developed I have found less and less of a need for these helper frameworks.		Prefer CoreData			iCarousel
2/27/2014 17:18:00	6 or more years	1 to 2 years	Network, Testing		AFNetworking, RestKit, JSONKit				Kiwi		ViewDeck/JASidePanels/ECSlidingViewController, DTCoreText
2/27/2014 17:43:12	6 or more years	6 or more years	Hardware addons	Hardware such as Bluetooth stylus, page turning foot pedals (for sheet music apps) etc. all require special frameworks. AFNetworking for all uri requests. I just easier and better to use.			SQLite	SQLite is a part of iOS, but I prefer it to CoreData.			
2/28/2014 7:42:53	3 to 5 years	1 to 2 years	Network, Social, Utilities	Social - Use facebook sdk Utilities - use sourcetree for git and kledascope for merge issues	AFNetworking, JSONKit, json-framework	See above	MagicalRecord	I wanted to see which one I liked better compared to just Core Data. Ended up using core data.		nope.	FlatUIKit, AwesomeMenu, iCarousel, REMenu
03/03/14 12:06	6 or more years	1 to 2 years									
3/28/2014 17:26:57	3 to 5 years	3 to 5 years	Network, Social, Utilities		AFNetworking, RestKit, SDWebImage, JSONKit, SDWebImage						
3/28/2014 19:50:24	3 to 5 years	3 to 5 years	Network	There are great networking libraries out there	AFNetworking	Because it's awesome?		Nope, just haven't used it.	Kiwi	Pretty syntax, mocking	
3/28/2014 22:47:05	3 to 5 years	3 to 5 years	Network, Data Store		AFNetworking, ASIHTTPRequest						
3/30/2014 2:54:47	3 to 5 years	1 to 2 years	Network, Debug	AFNetworking & MKNetworkKit really simplifies network handling For debug, I like using DLog for logging and Reveal to inspect the view hierarchy.	AFNetworking	See above	mogenerator	Easy to generate models for core data, and apply modifications	Kiwi	I like how Kiwi handles unit testing. Much more readable than Apple's ones.	ViewDeck/JASidePanels/ECSlidingViewController
04/01/14 07:06	6 or more years	3 to 5 years	Network, Data Store, Social, Utilities	Apple's frameworks are quite robust, but when it comes to things like network and data storage, there is just way too much 'boiler plate' code that frameworks can take care of.	AFNetworking	AFNetworking takes a lot of the hassle out of making simple GET and POST requests.	MagicalRecord	MagicalRecord makes Core Data code so much cleaner and easier to write, and I know the guys who maintains it :)		We don't do unit testing	
04/05/14 14:08	3 to 5 years	3 to 5 years	Network, Data Store, Social	Apple's own SDK either needs a lot of boiler plate code or just doesn't have the necessary tools for it (social).	AFNetworking, ASIHTTPRequest	These networking frameworks have a bit more functionality. AFNetworking also incorporates blocks instead of delegation. They're also supposedly faster than Apple's own.	MagicalRecord, mogenerator	mogenerator is pretty much inevitable because it creates subclasses for all of the objects in Core Data -- something that otherwise has to be done manually. MagicalRecord greatly eases the manipulation of data (fetching, creating etc).			ViewDeck/JASidePanels/ECSlidingViewController, iCarousel

