



آموزش برنامه نویسی اندروید در محیط اندروید استودیو

کار با وب سرویس در اندروید
بخش اول: وب سرویس چیست

مدرس : سید مهدی مطهری

www.android-studio.ir



به نام خدا

وب سرویس چیست؟

وب سرویس، استاندارد است برای انتقال داده بین پلتفرم‌ها / ماشین‌ها / نرم افزارهای مختلف (عموما بین سرویس دهنده و سرویس گیرنده یا همان سرور و کلاینت). این استاندارد در محیط های مختلف یکسان است و فارغ از نوع سخت افزار یا سیستم عامل یا زبان برنامه نویسی بکار رفته در نرم افزار مبدا و مقصد، به راحتی می‌توان داده ها را ارسال و دریافت کرد بطوری که این داده ها برای هردو طرف قابل فهم باشد. همانطور که از نام این استاندارد پیداست، وب سرویس‌ها تحت وب قابل دسترسی هستند و انتقال داده ها از طریق پروتکل‌های استاندارد مانند HTTP صورت می‌پذیرد.



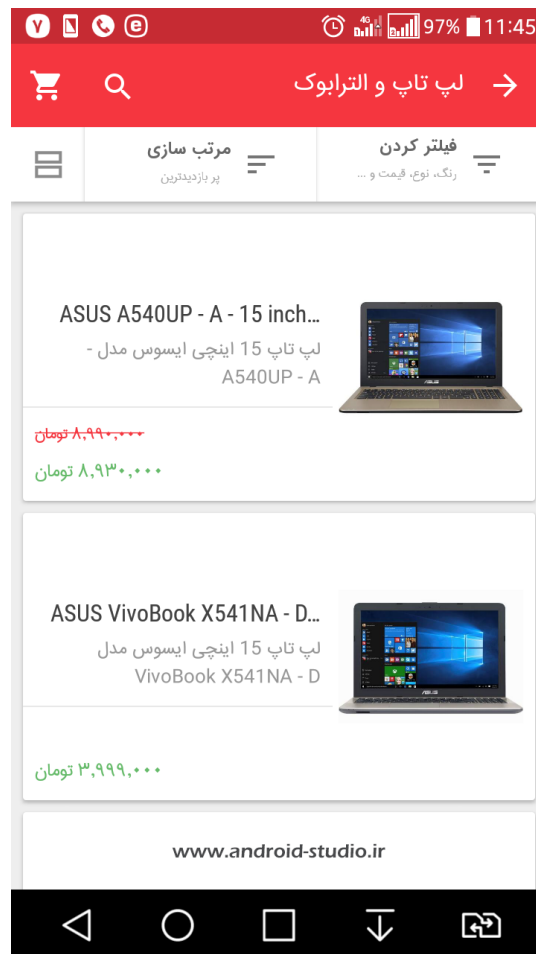


امروزه وب سرویس‌ها کاربرد فراوانی دارند. بخش زیادی از مردم از وب سرویس‌ها استفاده می‌کنند در حالی که خودشان متوجه این استفاده نمی‌شوند، به این دلیل که وب سرویس‌ها در پشت صحنه کار می‌کنند.

مثال‌های متعددی برای موارد کاربرد وب سرویس می‌توان ذکر کرد. یک فروشگاه اینترنتی را در نظر بگیرید. کاربر با مراجعه به وب سایت فروشگاه، به لیست محصولاتی که روی سرور این سایت قرار گرفته دسترسی دارد و می‌تواند جزئیات محصولات از جمله نام کالا، قیمت، رنگ و اندازه را مشاهده نماید. این اطلاعات در یک دیتابیس بر روی سرور ذخیره شده است. حالا مدیر این فروشگاه آنلاین قصد دارد یک اپلیکیشن نیز در اختیار کاربران قرار دهد تا کاربر بدون نیاز به مراجعه به وب سایت و تنها با نصب برنامه روی تلفن هوشمند یا تبلت خود بتواند به لیست محصولات موجود در سایت دسترسی داشته و در صورت نیاز، یک محصول را به صورت آنلاین خریداری کند. مسلماً نمی‌توان لیست و جزئیات محصولات را به صورت لوکال (محلی) بر روی دیتابیس داخلی اپلیکیشن وارد کرد زیرا داده‌های مرتبط با محصولات دائماً در حال تغییر هستند. هر روز تعدادی محصول جدید اضافه می‌شود، تعدادی محصول حذف می‌شود، قیمت‌ها مرتباً در حال تغییر هستند و... که اگر مدیر فروشگاه بخواهد تمام این تغییرات را درون دیتابیس داخلی اپلیکیشن اعمال کند، لازم است در هر روز چندین بار نسخه جدیدی از اپ فروشگاه را منتشر کند و کاربر هم مجبور است دائماً در حال بروزرسانی و دانلود و نصب نسخه‌های جدید نرم افزار باشد که عملاً این کار غیر ممکن و طاقت فرساست؛ هم برای سازنده اپلیکیشن و هم کاربر. علاوه بر اشکالات قبل، در صورت ثبت اطلاعات محصولات در دیتابیس داخلی اپلیکیشن، حجم دیتابیس و در نهایت حجم اپ شدیداً افزایش یافته و برای فروشگاه‌های بزرگ ممکن است به چند صد مگابایت یا چندین گیگابایت برسد! پس راه حل منطقی دریافت اطلاعات محصولات موردنظر کاربر به صورت آنلاین از سرور است. در اینجا وب سرویس به کمک مدیر فروشگاه می‌آید. با استفاده از وب سرویس به راحتی می‌توان اپلیکیشن را به سرور و دیتابیس وب سایت فروشگاه متصل کرد بدون آنکه نیاز به استفاده از سرور مجزا و مخصوصی باشد. یعنی مهم نیست سیستم عامل سرور این وب سایت، ویندوز است یا لینوکس، وب سایت با زبان PHP نوشته شده یا ASP، دیتابیس از نوع MySQL است یا Oracle یا SQL Server. هیچ محدودیتی نیست. در طرف کلاینت یا سرویس گیرنده نیز همینطور. یعنی مهم نیست دیوایس کاربر Android باشد یا iOS یا Windows؛ اپلیکیشن با زبان java نوشته شده یا kotlin یا swift یا C++. به هر روشی که سرور و کلاینت پیاده سازی شده باشند، امکان استفاده از وب سرویس یا به عبارتی API ها فراهم است. بنابراین یک فروشگاه آنلاین به راحتی می‌تواند کاربران دارای دیوایس‌های Android ، iOS ، Windows ، BlackBerry و... را بطور همزمان و بدون نیاز به بکارگیری سخت افزار و یا نرم افزار مجزا برای هرکدام از این محیط‌های کاربری، پوشش دهد. فرمتی که داده‌ها از سمت سرور ارسال می‌شود برای اپلیکیشن اندروید و اپلیکیشن ویندوز و سایر سیستم عامل‌ها یکسان است و هیچ تفاوتی نمی‌کند.



نکته دیگر اینکه در وب سرویس فقط داده ها ارسال می شوند و جزئیات و عناصر رابط کاربری، نحوه چیدمان اجزاء، رنگ ها و... ارتباطی به وب سرویس ندارد.



تصویر فوق مربوط به یک اپلیکیشن فروش آنلاین است. ابتدا کاربر دسته بندی محصول مدنظر خود را تعیین می کند. سپس این دسته بندی توسط وب سرویس از سرور وب سایت فروشگاه دریافت و در محیط اپلیکیشن برای کاربر نمایش داده می شود. نام لپ تاب، توضیحات، قیمت و آدرس (Url) تصویر مربوط به هر لپ تاب، مواردی هستند که توسط وب سرویس به کلاینت یا همان دیوایس کاربر ارسال گردیده است. مابقی اجزائی که در صفحه مشاهده می کنید مربوط به خود اپلیکیشن هستند و از سرور گرفته نشده است.

سامانه های پیامکی مثال دیگری از کاربرد وب سرویس ها هستند. از این سامانه ها جهت ارسال پیامک از طریق یک پنل مدیریتی تحت وب استفاده می شود و نیازی به سیم کارت و گوشی نیست. مالک سامانه می تواند به سادگی و از طریق پنل مدیریت تحت وب سامانه خود که آنرا از یک شرکت خریداری نموده، اقدام به ارسال پیامک به شماره های مختلف نماید.

اما این سامانه ها امکان دیگری نیز در اختیار مالک سامانه قرار می دهند. یکی از این امکانات API (مخفف عبارت Application Programming Interface) یا همان وب سرویس است که به کاربر اجازه



می‌دهد بدون نیاز به ورود به پنل مدیریتی سامانه (از طریق مرورگر) و بطور خودکار، پیامک ارسال کند. به عنوان مثال یک فروشگاه آنلاین در نظر دارد بعد از هر ثبت سفارش خرید کالا توسط مشتری، کد رهگیری برای وی در قالب پیامک ارسال شود. اینجا نیاز به ارسال دستی پیامک توسط کارمندان فروشگاه نیست بلکه تنها کافیه سیستم فروشگاه آنلاین به یک سامانه پیامکی که API ارائه می‌دهد متصل شود تا بعد از هر ثبت سفارش توسط مشتریان، یک درخواست ارسال پیامک به سامانه ارسال گردد. این درخواست شامل متن پیامک و شماره گیرنده است. در اینجا مهم نیست سرور فروشگاه لینوکسی است یا ویندوزی یا سیستم فروشگاه با چه زبانی نوشته شده. سامانه تنها یک درخواست با فرمت استاندارد را می‌پذیرد که شامل متن پیام و شماره مقصد است و اینکه این درخواست از سمت چه سروری و چه سیستمی ارسال شده تفاوتی نمی‌کند.

وبسایت‌های نمایش وضعیت آب و هوا نیز معمولا API در اختیار افراد قرار می‌دهند (این API می‌تواند رایگان و یا مشمول هزینه باشد). استفاده از این سرویس بسیار گسترده است. به عنوان مثال مدیر یک وبسایت یا یک اپلیکیشن می‌تواند با استفاده از این API، آب و هوای مربوط به شهرهای مختلف را درون وبسایت یا اپ خود نمایش دهد بدون اینکه لازم باشد شخصا اطلاعات وضعیت آب و هوای تک تک شهرها و کشورها را جمع‌آوری کند. کافیه از طرف وبسایت یا اپ مربوطه، نام شهر یا مختصات جغرافیایی آن از طریق وبسرویس به سرور وبسایت اصلی ارسال گردد تا در کمترین زمان ممکن پاسخ به سمت کلاینت ارسال گردد.

AccuWeather.com یکی از معروفترین سرویس‌های ارائه اطلاعات آب و هوایی جهان است.


<https://developer.accuweather.com/packages>

www.android-studio.ir

Packages

Limited Trial	Standard		Prime		Elite	
Locations	Locations		Locations		Locations	
Current Conditions	Current Conditions		Current Conditions		Current Conditions	
24-Hour Historical Current Conditions	24-Hour Historical Current Conditions		24-Hour Historical Current Conditions		24-Hour Historical Current Conditions	
Forecast 5-day	Forecast 5-day		Forecast 10-day		Forecast 15-day	
Forecast 12-hour	Forecast 12-hour		Forecast 72-hour		Forecast 120-hour	
Indices 5-day	Indices 5-day		Indices 10-day		Indices 15-day	
	Alarms 5-day		Alarms 10-day		Alarms 15-day	
	Translations		Translations		Translations	
			Tropical		Tropical	
			Alerts		Alerts	
			Imagery		Imagery	
	Standard Pricing		Prime Pricing		Elite Pricing	
50 calls/day Limit: 1 key/developer Free: Get started now!	Base Monthly Fee	CPM Rate	Base Monthly Fee	CPM Rate	Base Monthly Fee	CPM Rate
	\$25.00 per month	\$0.12 CPM over 225,000 calls per month	\$250.00 per month	\$0.15 CPM over 1,800,000 calls per month	\$500.00 per month	\$0.22 CPM over 2,400,000 calls per month
	Purchase Standard Package		Purchase Prime Package		Purchase Elite Package	

در انتهای وبسایت، گزینه ای با نام AccuWeather APIs قرار دارد که با کلیک روی آن به قسمتی با نام Developer منتقل می‌شویم. همانطور که از نام آن پیداست، این امکانات ویژه توسعه دهندگان است. سپس به جدولی می‌رسیم که هزینه دسترسی به API را در پلن‌های مختلف نمایش می‌دهد. به عنوان مثال برای ۵۰ درخواست در روز، هزینه ای دریافت نمی‌شود و استفاده از آن رایگان است. یعنی توسعه دهنده می‌تواند با استفاده از این وب سرویس، به تعداد ۵۰ بار در روز، وضعیت آب و هوایی شهرهای مختلف را در اپلیکیشن یا وبسایت خود نمایش دهد؛ بدون پرداخت هزینه. اما برای استفاده‌های بالاتر باید هزینه متناسب با آن را پرداخت کرد.

تذکر: مفهوم API با Web Service برابر نیست و در واقع وب سرویس به نوعی زیرمجموعه API محسوب می‌شود. اما عمدتاً از واژه API در سرویس‌هایی مانند سامانه های پیامکی یا سایت های آب و هوایی و... استفاده می‌گردد.

سیستم‌های رزرو بلیت را به عنوان آخرین مثال از کاربردهای وب سرویس بررسی می‌کنم. مطمئناً اپلیکیشن‌ها و یا وبسایت‌های متعددی را در ذهن دارید که خدمات رزرو و خرید انواع بلیت‌های هواپیما، قطار و اتوبوس ارائه می‌دهند. شاید این سوال در ذهن شما بوجود آمده که آیا همه این سرویس دهنده‌ها به صورت جداگانه با چند صد شرکت مربوط به حمل و نقل هوایی، ریلی و جاده ای قرارداد



بسته اند؟ پاسخ منفی است! تعداد سرویس دهنده هایی که به صورت مستقیم با شرکت های مسافری قرارداد امضا کرده و در تعامل هستند، کمتر از انگشتان یک دست است. مابقی ارائه دهندگان خدمات خرید بلیت در قالب دریافت نمایندگی از بستر این شرکت ها استفاده می کنند. به اینصورت که با استفاده از API یی که شرکت مادر در اختیار سرویس دهندگان زیرمجموعه خود قرار می دهد، می توانند در اپلیکیشن یا وبسایت خود بدون اینکه نامی از شرکت مادر باشد، اقدام به فروش بلیت نموده و کسب درآمد کنند. در واقع این سرویس ها تنها یک واسطه بین شرکت مادر (طرف قرارداد با شرکت های مسافری) و مشتری نهایی هستند و از هر خریدی که توسط اپلیکیشن یا وبسایت آنها انجام می شود، پورسانتی از شرکت مادر دریافت می کنند.

مثالهای پراکنده و گسترده ای را مطرح کردم تا دید شما نسبت به کاربرد وب سرویس ها کاملاً باز شود. اما کاربرد وب سرویس می تواند برخلاف موارد فوق، محدود و بسیار ساده باشد. فرض کنید می خواهیم اپلیکیشنی بسازیم که افراد از طریق آن بتوانند از خدمات کارواش در محل بهره مند شوند. ساده ترین حالت برای پیاده سازی این اپ به اینصورت است که کاربر پس از نصب برنامه روی دیوایس اندرویدی خود ابتدا فرم ثبت نام را تکمیل کرده و سپس با ذکر آدرس محل سکونت یا مشخص کردن آن روی نقشه، درخواست ارسال کارواش سیار نماید. در این مثال، داده هایی که بین دیوایس (کلاينت) و سرور رد و بدل میشود مربوط به مشخصات فردی شخص و محل درخواست کارواش است. این سرور را خود توسعه دهنده می تواند با پرداخت هزینه اندکی راه اندازی نماید و مانند مثال های قبل، سرویس دهنده واسطی درکار نیست تا بتوانیم از API آن استفاده کنیم. بنابراین راه اندازی وب سرویس به عهده خود ماست.

در نهایت اگر بخواهیم تفاوت یک Web Service را با یک Web Page (صفحات وب) بررسی کنیم، در Web Page که کاربر توسط یک مرورگر آنرا مشاهده می کند، تمامی اجزای صفحه از جمله متن، تصاویر، رنگها، چینش عناصر و... همگی از سمت سرور و با فرمت HTML دریافت می شوند اما در Web Service تنها محتوای اصلی از سرور دریافت شده و در چارچوبی که درون اپلیکیشن برای نمایش محتوا از قبل طراحی شده، به کاربر نمایش داده می شود. بطور خلاصه در یک صفحه وب، قالب (رابط کاربری) نیز از طرف سرور دریافت می شود اما در اپلیکیشنی که از وب سرویس استفاده می کند، رابط کاربری درون اپلیکیشن قرار گرفته و تنها محتوا است که توسط وب سرویس دریافت شده و درون این رابط کاربری قرار می گیرد.



وب سرویس RESTful:

{ REST }

در حال حاضر دو وب سرویس SOAP (مخفف Simple Object Access Protocol) و REST (مخفف Representational State Transfer) بیشترین کاربرد را در بین توسعه دهندگان دارند. اخیراً و با توجه به سادگی در پیاده سازی وب سرویس REST، این گزینه طرفداران بیشتری به خود جذب کرده است. البته یک تفاوت اساسی بین این دو وجود دارد. SOAP یک پروتکل بوده درحالی که REST یک سبک معماری وب سرویس است که از طریق پروتکل HTTP داده ها را ارسال و دریافت می کند.

در اندروید نیز از معماری REST استفاده می کنیم. وب سرویس هایی که توسط معماری REST پیاده سازی می شوند را RESTful می نامند.

در ابتدای مبحث راجع به فرمت انتقال داده در وب سرویس صحبت شد و اینکه این فرمت در همه پلتفرم ها و زبان های رایج برنامه نویسی پشتیبانی می شود. SOAP تنها از فرمت XML برای انتقال داده استفاده می کند درحالی که در وب سرویس های RESTful از سایر فرمت ها مانند HTML و JSON نیز پشتیبانی می شود که در این بین، فرمت JSON به علت مصرف کمتر پهنای باند، سرعت بیشتری نسبت به سایر گزینه ها دارد. نتیجه آن است که در اندروید نیز از فرمت JSON برای ارتباط بین سرور و کلاینت استفاده می شود. ارسال و دریافت اطلاعات توسط متدهایی مانند GET، POST، PUT و DELETE برای عملیاتی مانند دریافت، ارسال، ویرایش، حذف و... انجام می گردد.

مانند سایر قسمت های اندروید، در مبحث وب سرویس و اتصال اپلیکیشن به سرور نیز کتابخانه های متعددی منتشر شده و نیاز ما به درگیر شدن با مفاهیم و مسائل فنی و پیچیده ی پیاده سازی وب



سرویس‌ها را مرتفع می‌سازد. در حال حاضر دو کتابخانه **Retrofit** و **Volley** در بین توسعه دهندگان اندروید بیشترین کاربرد را دارند. کتابخانه Retrofit توسط شرکت Square و کتابخانه Volley توسط Google منتشر شده و پشتیبانی می‌شوند. در مبحث بعد به بررسی و نحوه کار با کتابخانه Retrofit خواهیم پرداخت.

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش های بهتر یاری فرمائید.

www.android-studio.ir