

بسمه تعالی



دانشگاه جامع علمی کاربردی

رابط برنامه نویسی برنامه

API (Application Programming Interface)



مدرس: مهندس نوید همراهی

n.hamrahi@gmail.com

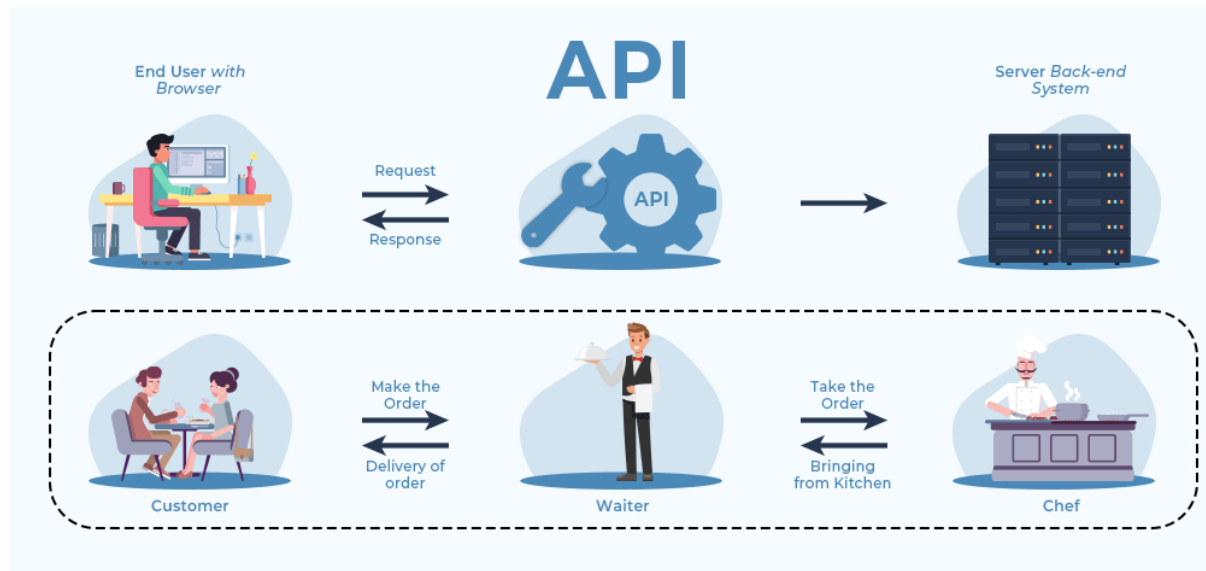
www.navidhamrahi.ir

جهت دریافت جزوه و اطلاع از روند کلاس به وب سایت بالا

بخش « دانشگاه مجازی » مراجعه فرمایید.

API چیست ؟

در این آموزش قرار است که بگوییم API چیست. API مخفف عبارت "رابط برنامه نویسی اپلیکیشن" است. در واقع مجموعه ای از دستورات و قوانین در برنامه نویسی است که توسط توسعه دهندگان نرم افزار برای ارتباط و تعامل با یک برنامه کامپیوتری یا سیستم ایجاد خواهد شد.



به طور کلی API به توسعه دهندگان اجازه می دهد تا اطلاعات را بین برنامه ها منتقل کنند، عملیات مختلفی را انجام دهند و به منابع سیستم دسترسی داشته باشند. در ادامه می گوییم API چیست و راجع به انواع، کاربردها، مزایا و معایب آن بیشتر صحبت خواهیم کرد.

Application Programming Interface چیست؟

رابط برنامه نویسی اپلیکیشن یا به اختصار API، یک مجموعه شناخته شده از دستورات و قوانین است که به توسعه دهندگان اجازه می دهد تا با سیستم های نرم افزاری دیگر تعامل کنند و از ویژگی ها و داده های آنها بهره ببرند.

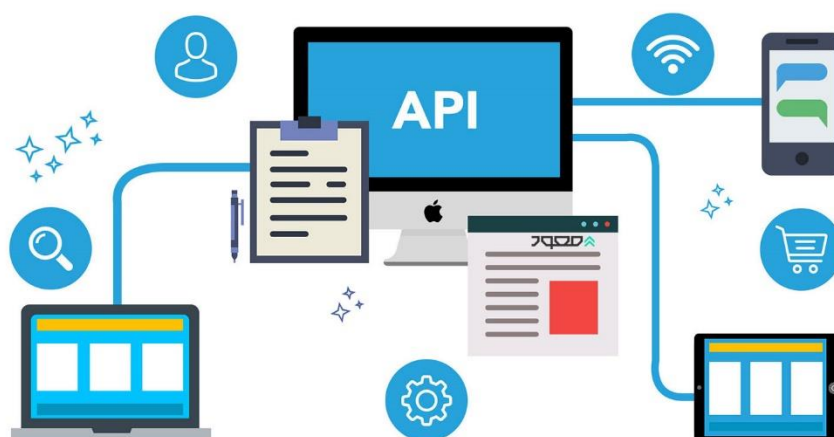
این به توسعه‌دهندگان امکان متصل کردن و تعامل با برنامه‌ها و سرویس‌های مختلف را داده است، بدون آن که نیاز به دسترسی به کد منبع برنامه داشته باشند. این نکته را بدانید که APIها در توسعه نرم‌افزارهای متعددی از وبسایت‌ها تا اپلیکیشن‌های تلفن همراه و حتی دستگاه‌های خانگی مورد استفاده قرار می‌گیرند.

به عنوان مثال Google Maps یک API ساده در استفاده از نقشه‌های آنلاین است. توسعه‌دهندگان می‌توانند API Google Maps را در وبسایت یا اپلیکیشن خود به کار ببرند تا نقشه‌های تعاملی بسازند یا اطلاعات مکانی مشخصی را به کاربران ارائه دهند.

این API امکاناتی مانند نمایش نقشه، جستجوی مکان‌ها، مسیریابی و حتی ترافیک زنده را فراهم می‌کند. توسعه‌دهندگان از دستورات API برای ارسال درخواست‌ها به سرور Google Maps استفاده می‌کنند و سپس نتایج را در وبسایت یا اپلیکیشن خود نمایش می‌دهند.

تاریخچه API

تاریخچه API یا رابط برنامه‌نویسی اپلیکیشن به یکی از بنیان‌های حیاتی توسعه نرم‌افزار تبدیل شده است. این مفهوم از دهه ۱۹۵۰ میلادی شکل گرفت و در طول زمان تکامل یافت. در اوایل توسعه کامپیوترها، تمرکز کامل بر روی برنامه‌نویسی برای یک سیستم خاص بود و ارتباط بین برنامه‌ها به حداقل می‌رسید.



اما در دهه‌های بعد، با ظهور کامپیوترهای شخصی و سیستم‌های عامل، رابط‌های برنامه‌نویسی به وجود آمدند تا امکان استفاده از برنامه‌ها و دستورات خط فرمان را به خوبی فراهم کنند. سپس با توسعه سیستم‌های توزیع شده و اینترنت در دهه ۱۹۹۰، نیاز به راه‌حل‌های ارتباطی بین سیستم‌ها افزایش یافت.

CORBA و DCOM از جمله فناوری‌های معروف آن دوره خاص حساب می‌شدند. نهایتاً و در دهه ۲۰۰۰ و با ظهور وب و توسعه اپلیکیشن‌های تلفن همراه، REST و GraphQL به عنوان رویکردهای اصولی برای ایجاد و استفاده از API در آن سال معرفی شدند.

API چگونه کار می‌کند؟

به طور ساده، یک API دارای دستورات و متدهایی است که تعیین می‌کند چگونه یک برنامه می‌تواند با یک سیستم دیگر ارتباط برقرار کند. این متدها به برنامه‌نویسان امکان می‌دهند تا درخواست‌ها را به سیستم مبدأ ارسال کنند و اطلاعات مورد نیاز خود را از سیستم مقصد دریافت کنند. API‌ها معمولاً مستندات دارند که توسط سازندگان آن تهیه می‌شوند.



این مستندات شامل توضیح دستورات موجود، پارامترهای لازم، و نحوه استفاده از API هستند که به برنامه‌نویسان کمک می‌کنند تا آن را به درستی به کار ببرند. بدانید که، زمانی که یک برنامه استفاده از یک API را می‌خواهد، درخواست‌های HTTP یا سایر پروتکل‌های ارتباطی مشابه به سرور می‌فرستد.

سرور سپس این درخواستها را پردازش کرده و پاسخی را باز می‌گرداند که شامل اطلاعات یا عملیات مورد نظر است. این پاسخ سپس توسط برنامه مشتری (مثلاً یک اپلیکیشن تلفن همراه) پردازش می‌شود. با این روند، برنامه‌ها می‌توانند به طور مؤثر با یکدیگر تعامل کنند و از خدمات و اطلاعات موجود در سیستم‌های دیگر بهره برداری کنند.

چرا به رابط برنامه‌نویسی کاربردی نیاز داریم؟



دلایل مهمی برای نیاز به API عبارتند از:

تعامل و اشتراک داده: امروزه برنامه‌ها و سیستم‌ها نیاز به تعامل و اشتراک داده دارند. API این امکان را فراهم می‌کند تا برنامه‌ها بتوانند اطلاعات را با یکدیگر به اشتراک بگذارند. مثلاً یک برنامه می‌تواند از API یک سرویس نقشه استفاده کند تا مکان کاربر را نمایش دهد.

توسعه تخصصی: با استفاده از API، توسعه‌دهندگان می‌توانند برنامه‌های تخصصی و قدرتمندترین ایجاد کنند. آنها نیازی به نوشتن کد از ابتدا ندارند و می‌توانند از قابلیت‌ها و خدمات دیگران بهره‌برند.

سهولت توسعه و تغییرات: API از تغییرات در دیگر بخش‌های برنامه محافظت می‌کند. اگر یک بخش از برنامه تغییر کند، API می‌تواند تغییرات را به سایر برنامه‌ها اعلام کند تا سازگاری حفظ شود.

توسعه چند پلتفرمی: با استفاده از API، می‌توان برنامه‌ها را برای چند پلتفرم مختلف ایجاد کرد. به عنوان مثال، یک سرویس وب می‌تواند API خود را ارائه دهد تا برنامه‌های تلفن همراه، وب، و دسکتاپ از آن استفاده کنند.

امنیت: API به توسعه‌دهندگان امکان مدیریت دقیق دسترسی به منابع و اطلاعات را می‌دهد. این اجازه می‌دهد تا سیاست‌های امنیتی به صورت مرکزی اعمال شوند.

کاربردهای API

۱- آسان‌تر کردن کارها برای توسعه‌دهندگان

با استفاده از API، توسعه‌دهندگان نیازی به نوشتن کدهای اصلی برای ایجاد عملکردهای پایه‌ای ندارند. آنها می‌توانند از عملکردهای آماده و متعددی که API ارائه می‌کند بهره‌برداری کنند. این امر توسعه نرم‌افزار را به شدت تسریع می‌دهد.

همچنین API به توسعه‌دهندگان این امکان را می‌دهد که کدهایی که ممکن است در برنامه‌های مختلف تکرار شوند را به یک مکان مشترک انتقال دهند. این کاهش تکرار کدها، به معنای افزایش کارایی و حذف اشکالات احتمالی است. API اغلب از استانداردهای خاصی پیروی می‌کند که توسعه‌دهندگان می‌توانند به عنوان یک راهنمای کاربردی استفاده کنند.

این استانداردها به ایجاد یک رویکرد یکنواخت برای توسعه کمک می‌کنند و از تداخل‌های احتمالی جلوگیری می‌کنند و در آخر توسعه‌دهندگان می‌توانند به راحتی به خدمات و داده‌های خارجی دسترسی پیدا کنند. در کل این امر به توسعه‌دهندگان این امکان را می‌دهد که از منابع و اطلاعات بیرونی بهره‌برداری کرده و امکانات خود را به طور چشم‌گیری افزایش دهند.

۲- کنترل دسترسی به منابع

کنترل دسترسی به منابع یکی از جنبه‌های بسیار حیاتی و اساسی در استفاده از رابط برنامه‌نویسی کاربردی است. این مفهوم به توسعه‌دهندگان اجازه می‌دهد تا بطور دقیق مشخص کنند که چه کسانی به منابع و خدمات ارائه شده توسط یک API دسترسی دارند و چه کسانی ندارند.

در واقع کنترل دسترسی به منابع از اهمیت بسیار بالایی در حفظ امنیت سیستم‌ها و داده‌ها برخوردار است. با استفاده از مکانیسم‌های کنترل دسترسی، می‌توان جلوی دسترسی غیرمجاز به منابع حیاتی را گرفت. کنترل دسترسی به منابع به تعیین نقش‌ها و مجوزهای کاربران بستگی دارد. توسعه‌دهندگان می‌توانند نقش‌ها مختلفی مانند مدیران، کاربران عادی، یا مهمانان تعریف کنند و به هر نقش مجوزهای مشخصی اختصاص دهند.

این قابلیت به مدیران امکان می‌دهد تا دقیقاً تعیین کنند که کدام کاربران یا برنامه‌ها به منابعی دسترسی دارند و تا چه حد. به عنوان مثال، ممکن است بخواهند تعیین کنند کدام کاربران توانایی ایجاد، حذف، یا تغییر داده‌ها را دارند.

۳- ارتباط بین خدمات

API به توسعه‌دهندگان این امکان را می‌دهد تا سیستم‌های جدید را با استفاده از خدمات و منابع موجود ایجاد کنند. این انعطاف‌پذیری امکان توسعه سریع و کارآمد را فراهم می‌کند.

با استفاده از ارتباط بین خدمات، سیستم‌ها می‌توانند وظایف مختلف را بر روی خدمات مجزا تقسیم کنند. این تقسیم کار به مدیریت منابع بهینه کمک می‌کند و امکان پیشرفت همزمان در چندین بخش را فراهم می‌سازد. ارتباط بین خدمات اجازه توزیع بار کاری بین خدمات مختلف را می‌دهد. این امکان به بهره‌برداری بهینه از منابع سروری و جلوگیری از افت کارایی در مواجهه با بارهای زیاد کمک می‌کند.

با استفاده از API، می‌توان اطلاعاتی مانند عملکرد خدمات و داده‌ها را به صورت زنده رصد کرده و مدیریت کرد. این امکان به افزایش کارایی و امنیت سیستم‌ها کمک می‌کند.

به طور خلاصه، ارتباط بین خدمات با استفاده از API یک اصل اساسی در توسعه نرم افزار و ایجاد سیستم های پیچیده و هماهنگ است. این امکان به توسعه دهندگان این قدرت را می دهد که سیستم ها و خدمات را با انعطاف و کارایی بالا ایجاد و مدیریت کنند.

۴- امنیت بیشتر

یکی از اصول امنیتی اصلی در API احراز هویت کاربران و اعطای دسترسی های مورد نیاز به آنها است. این امر از طریق مکانیسم های مانند توکن های امنیتی (Security Tokens) و API Key انجام می شود. برای حفاظت از داده های حساسی که از طریق API منتقل می شوند، رمزنگاری داده (Data Encryption) استفاده می شود.

این رمزنگاری باعث می شود تا حتی اگر داده ها در مسیر ارتباطی انتقال یابند، برای اشخاص غیرمجاز غیرقابل فهم باقی بمانند. همچنین API به مدیران امکان می دهد تا دقیقاً تعیین کنند کدام کاربران یا برنامه ها به چه منابع و خدماتی دسترسی دارند.

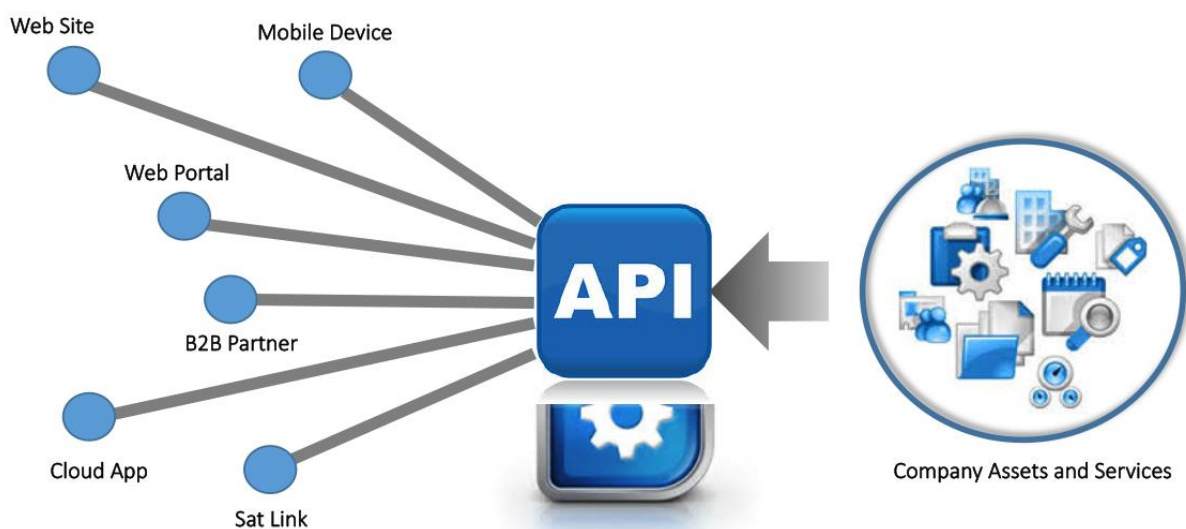
این مدیریت دسترسی ها به جلوگیری از دسترسی های غیرمجاز کمک می کند. این نکته را بدانید که به وسیله ای ابزارهای مانیتورینگ و رصد، می توان فعالیت های API را پیگیری کرده و هرگونه ناهنجاری یا تلاش برای نفوذ را تشخیص داد. این مانیتورینگ به سرعت واکنش به وقوع رخ داده های امنیتی کمک می کند.

۵- پیشرفت و توسعه عملکرد

با استفاده از API، توسعه دهندگان می توانند به سرعت و به راحتی امکانات جدید به برنامه ها و سیستم ها اضافه کنند. این امر بهبود عملکرد و افزایش توانمندی های نرم افزار را ممکن می سازد. API امکان اتصال به سرویس ها و منابع خارجی را به توسعه دهندگان می دهد. این ارتباط با سرویس های خارجی می تواند به بهبود عملکرد و افزایش توانایی های برنامه ها منجر شود.

همچنین API به چندین برنامه یا سیستم اجازه می‌دهد تا از یک منبع یا خدمت مشترک بهره‌برند. این منابع مشترک می‌توانند در توسعه و بهبود عملکرد مشترک مؤثر باشند. توسعه‌دهندگان می‌توانند با استفاده از API به تحلیل و بهبود عملکرد برنامه‌ها پردازند.

این تحلیل‌ها می‌تواند به بهینه‌سازی فرآیندها و کاهش زمان اجرا کمک کند. با API، می‌توان از منابع ابری مثل سرورها، ذخیره‌سازی ابری، و خدمات مرتبط بهره‌برداری کرد. این امکان به توسعه‌دهندگان امکان دسترسی به زیرساخت‌های پیچیده را فراهم می‌کند.



مزایای استفاده از API

۱- بهره‌وری

بهره‌وری یکی از مزایای کلیدی استفاده از رابط برنامه‌نویسی کاربردی است. API به برنامه‌ها اجازه می‌دهد تا از منابع و خدمات مشترک بهره‌برند. این به اشتراک‌گذاری منابع می‌تواند بهبود بهره‌وری در زمینه مصرف منابع سیستمی و اقتصادی‌ترین استفاده از آنها را فراهم کند.

با استفاده از API، توسعه‌دهندگان می‌توانند قطعات کدهای آماده را به برنامه‌ها خود اضافه کنند، که منجر به کاهش زمان توسعه و هزینه‌های مرتبط با آن می‌شود.

۲- ارائه خدمات انعطاف پذیر

با API، می‌توان چندین سرویس و منبع مختلف را ترکیب کرد و خدمات جدیدی را ایجاد کرد. این ترکیب‌ها می‌توانند به کاربران امکانات بیشتر و متنوع‌تری ارائه دهند. همچنین API به کسب‌وکارها این امکان را می‌دهد تا به راحتی در بازارهای جدید عرضه کنند.

آنها می‌توانند با ارائه API به توسعه‌دهندگان و شرکت‌های شخص ثالث اجازه دسترسی به سیستم‌ها و خدمات خود را بدهند و درآمد جدیدی را تولید کنند.

۳- ادغام

API به توسعه‌دهندگان این امکان را می‌دهد تا داده‌های مختلف را از منابع مختلف به دست آورده و در یک مکان مشترک جمع‌آوری کنند. این ادغام داده‌ها می‌تواند به تجزیه و تحلیل دقیقتر و تصمیم‌گیری‌های بهتر کمک کند.

با API، می‌توان خدمات مختلف را به یکدیگر متصل کرده و خدمات جدیدی را ارائه داد. به عنوان مثال، ادغام یک سیستم پرداخت با یک فروشگاه آنلاین می‌تواند به مشتریان امکان پرداخت آسان‌تری را فراهم کند.

۴- اتوماسیون

انسان‌ها به اشتباه ممکن است خطاها را وارد کنند، اما اتوماسیون تکراری و بدون اشتباه فرآیندها را انجام می‌دهد، که به کاهش خطاها و بهبود دقت کمک می‌کند.

اتوماسیون به توسعه‌دهندگان و کسب‌وکارها این امکان را می‌دهد تا فرآیندها و وظایف تکراری را به سرعت و با کمترین تلاش انجام دهند. این به افزایش بهره‌وری و صرفه‌جویی در زمان و هزینه‌ها منجر می‌شود.

۵- عملکرد جدید

با ارائه عملکردهای جدید از طریق API، می‌توان به نیازهای متغیر بازار پاسخ داد. این امکان به شرکت‌ها این امکان را می‌دهد تا با رقابت در بازار رقابتی و متغیر همگام شوند.

با ارائه API، شرکت‌ها می‌توانند به توسعه‌دهندگان شخص ثالث اجازه دسترسی به سیستم‌ها و خدمات خود را بدهند. این توسعه‌دهندگان می‌توانند افزونه‌ها و افزودنی‌های جدیدی را ایجاد کرده و درآمد از آنها کسب کنند.

معایب استفاده از API

۱- هزینه

ایجاد یک API و اضافه کردن آن به سیستم ممکن است نیاز به منابع مالی و انسانی برای توسعه و پیاده‌سازی داشته باشد. این مراحل ابتدایی ممکن است هزینه‌بر باشند و نیاز به برنامه‌ریزی دقیق و تخصیص منابع مالی داشته باشند.

یکی از چالش‌های استفاده از API، هزینه نگهداری و به‌روزرسانی آن است. زمانی که یک API در حال استفاده است، نیاز به مراقبت و پشتیبانی دارد تا به درستی کار کند.

۲- مسائل امنیتی

کنترل دسترسی به منابع و عملیات API می‌تواند چالش‌های امنیتی ایجاد کند. اگر کنترل دسترسی به درستی انجام نشود، افراد غیرمجاز ممکن است به اطلاعات حساس دسترسی پیدا کنند.

در صورتی که API به اشتراک گذاشته شود یا داده‌های حساس از آن منتقل شود، ممکن است مسائل حریم خصوصی پیش بیاید. اطلاعات شخصی کاربران ممکن است در معرض خطر قرار گیرند، بنابراین باید مدیریت و حفاظت از حریم خصوصی به دقت انجام شود.



انواع API

در این بخش قرار است تا در مورد انواع API صحبت کنیم. API به سه بخش اصلی خصوصی، شراکتی و عمومی تقسیم می شوند. نوع خصوصی که بیشتر برای سازمان های خاصی تعریف می شود و شراکتی و عمومی به ترتیب برای چند سازمان مشترک و عموم توسعه دهندگان قابل استفاده است.

انواع API بر حسب نوع دسترسی

۱- API خصوصی

API خصوصی یک نوع رابط برنامه نویسی کاربردی (Application Programming Interface) است که مخصوص یک شرکت یا سازمان خاص تعریف و ارائه می شود و فریم ورک مخصوص خودش را دارد.

به دیگر اشیاء و نرم افزارهای داخلی آن سازمان اجازه ارتباط و تعامل با سیستمها و منابع خود را می دهد. این نوع API به صورت خصوصی و اختصاصی برای مصرف داخلی طراحی می شود و به عنوان امنیتی برای محافظت از منابع و داده های حساس سازمان استفاده می شود.

API خصوصی دارای کنترل دقیقی بر روی دسترسی به منابع و عملیات سیستم است. این به معنایی که تنها کاربران و اشیاء مجاز دسترسی به آن دارند و دیگران اجازه ندارند.

۲- API شراکتی

API شراکتی یک نوع رابط برنامه‌نویسی کاربردی (Application Programming Interface) است که بین دو یا چند سازمان یا سرویس مختلف به اشتراک گذاشته می‌شود تا اجازه تعامل و تبادل داده‌ها و خدمات بین آن‌ها را فراهم کند.

این نوع API به شرکت‌ها امکان اتصال و اشتراک منابع و خدمات خود با دیگران را می‌دهد تا به بهبود تجربه کاربری و افزایش بهره‌وری برسند. این نوع API معمولاً در محیط‌های کسب‌وکار و صنعتی به کار می‌رود و از اهمیت بالایی برخوردار است. API شراکتی به شرکت‌ها امکان ارتباط و تعامل با دیگر سازمان‌ها و خدمات را می‌دهد.

۳- API عمومی

API عمومی (Public API) یک نوع رابط برنامه‌نویسی کاربردی (Application Programming Interface) است که به صورت علنی و برای عموم توسعه‌دهندگان و استفاده عمومی در دسترس است.

این نوع API به سازمان‌ها، سرویس‌ها و سیستم‌های مختلف اجازه می‌دهد تا با دیگر سازمان‌ها و نرم‌افزارها به صورت مستقل و بدون نیاز به توافقات خاصی تعامل کنند.

API عمومی به عنوان یک واسط استاندارد برای دسترسی به سرویس‌ها و داده‌ها در اینترنت به کار می‌رود و در جهان دیجیتال به اهمیت بسیاری دست پیدا کرده است.

انواع API عمومی

رایگان

API رایگان یک نوع رابط برنامه‌نویسی کاربردی (Application Programming Interface) است که برای استفاده عموم و بدون هزینه به توسعه‌دهندگان و کاربران عرضه می‌شود.

این نوع API بدون محدودیت از نظر مالی ارائه می‌شود و اغلب توسط سازمان‌ها یا سرویس‌ها به منظور تشویق توسعه‌دهندگان و جلب مخاطبین بیشتر به کار می‌رود. این API‌ها معمولاً به صورت محدودیت‌هایی ارائه می‌شوند که به تعداد درخواست‌ها یا مصرف داده‌ها اشاره دارند.

تجاری

API تجاری یک نوع رابط برنامه‌نویسی کاربردی (Application Programming Interface) است که به عنوان یک محصول یا خدمت از طرف یک سازمان یا شرکت تجاری به توسعه‌دهندگان یا سایر سازمان‌ها ارائه می‌شود.

API تجاری معمولاً با هزینه ارائه می‌شود و می‌تواند یک منبع درآمد برای سازمان تامین کننده باشد. توسعه‌دهندگان برای دسترسی به این API باید هزینه‌هایی را پرداخت کنند، سپس سازمان از درآمد حاصل از این API بهره‌مند می‌شود.

دسته‌بندی API براساس نوع استفاده

API‌ها براساس نوع استفاده‌شان می‌توانند به چند دسته تقسیم شوند. هر کدام از این دسته‌بندی‌ها نیازمندی‌های مختلفی دارند و توسعه‌دهندگان می‌توانند بر اساس نیاز خود از API‌های مختلف استفاده کنند تا ویژگی‌ها و عملکرد مورد نظر خود را در برنامه‌ها و سرویس‌های خود پیاده‌سازی کنند.

۱- API پایگاه داده (Database)

API پایگاه داده باید توانایی اتصال و ارتباط با متنوعی از پایگاه داده‌ها را داشته باشد، از جمله پایگاه داده‌های SQL مانند MySQL و PostgreSQL و یا پایگاه داده‌های NoSQL مانند MongoDB و Cassandra.

همچنین API پایگاه داده باید امکان کنترل دقیق دسترسی به داده‌ها را ارائه دهد. این به معنایی است که توسعه‌دهندگان می‌توانند تعیین کنند که کدام قسمت‌ها از داده‌ها قابل دسترسی هستند و چه عملیات‌هایی می‌توانند انجام دهند.

این نکته را بدانید که امنیت داده‌ها در پایگاه داده امری حیاتی است. API باید از تکنیک‌ها و استانداردهای امنیتی برای حفاظت از داده‌ها در هنگام انتقال و ذخیره سازی آنها استفاده کند و API باید امکان اجرای پرس‌وجوهای پیچیده و فیلترینگ داده‌ها را از طریق پایگاه داده فراهم کند تا توسعه‌دهندگان بتوانند به داده‌ها بر اساس شرایط خاص دسترسی داشته باشند.

۲- API سیستم‌عامل (OS)

API سیستم‌عامل دارای ویژگی‌ها و عملکردهای متعددی است که به توسعه‌دهندگان در توسعه نرم‌افزارهای متنوع کمک می‌کند. این ویژگی‌ها عبارتند از:

مدیریت منابع سخت‌افزاری: این API به توسعه‌دهندگان امکان مدیریت و کنترل منابع سخت‌افزاری نظیر پردازنده، حافظه، دیسک‌ها، و دیگر دستگاه‌ها را ارائه می‌دهد. این امکان بهینه‌سازی استفاده از منابع و افزایش عملکرد سیستم را فراهم می‌کند.

مدیریت فایل‌ها و دستگاه‌ها: API سیستم‌عامل به توسعه‌دهندگان اجازه می‌دهد تا با فایل‌ها و دستگاه‌های ورودی و خروجی نظیر پوشه‌ها، فایل‌ها، چاپگرها، و دستگاه‌های شبکه تعامل کنند.

تعامل با شبکه: این API به برنامه‌ها امکان اتصال به شبکه‌های مختلف و ارسال و دریافت داده از این شبکه‌ها را فراهم می‌کند. این امکان به توسعه‌دهندگان ایجاد برنامه‌هایی که به اینترنت و شبکه‌های محلی متصل هستند را ممکن می‌سازد.

۳- API راه دور (Remote)

API راه دور معمولاً از پروتکل‌های شبکه مانند HTTP، HTTPS، FTP و غیره برای ارسال درخواست‌ها و دریافت پاسخ‌ها استفاده می‌کند. این API ها می‌توانند برای مختلف موارد مانند دسترسی به اطلاعات از پایگاه داده‌های ابری، انجام عملیات از راه دور بر روی سرورها، یا ارسال و دریافت داده‌ها بین دستگاه‌های مختلف به کار روند.

ویژگی‌ها و مزایای API راه دور به شرح زیر است:

دسترسی از هر نقطه: API راه دور به توسعه‌دهندگان این امکان را می‌دهد که از هر نقطه‌ای که به اینترنت متصل هستند، به منابع و خدمات دسترسی پیدا کنند. این امکان به اونیت‌های مختلف و کاربران این اجازه را می‌دهد که به منابع مشترک از راه دور دسترسی داشته باشند.

افزایش انعطاف‌پذیری: API راه دور به سازمان‌ها و توسعه‌دهندگان این امکان را می‌دهد تا به سرعت از منابع جدید بهره‌برداری کنند و به تغییرات در نیازها و تقاضاها پاسخ دهند.

۴- API وب (Web)

در زیر به ویژگی‌های API وب اشاره خواهیم نمود:

دسترسی به سرویس‌های آنلاین: با استفاده از API وب، توسعه‌دهندگان می‌توانند به سرویس‌ها و منابع آنلاین از جمله اطلاعات و داده‌های مختلف، خدمات وب، و اپلیکیشن‌های آنلاین دسترسی داشته باشند.

تبادل داده: API وب به توسعه‌دهندگان این امکان را می‌دهد تا داده‌ها را از سرویس‌ها و منابع وبی دریافت و به آنها داده ارسال کنند. این داده‌ها معمولاً به صورت JSON یا XML ارسال و دریافت می‌شوند.

انجام عملیات مختلف: API وب به توسعه‌دهندگان این امکان را می‌دهد تا عملیات‌های متنوعی از جمله جستجو، ایجاد، به‌روزرسانی، حذف داده‌ها و خدمات وب را انجام دهند.

انتقال فایل و رسانه: این API ها می‌توانند فایل‌ها و رسانه‌های مختلف نظیر تصاویر، ویدئوها، و سندها را برای اشتراک‌گذاری و دریافت ارسال کنند.