chatgpt

API یا Application Programming Interface مفهومی حیاتی در برنامه نویسی است که ارتباط و تعامل بین سیستم های نرم افزاری مختلف را تسهیل می کند. اساساً یک API روش ها و فرمت های داده ای را که برنامه ها می توانند برای درخواست و تبادل اطلاعات با یکدیگر استفاده کنند، تعریف می کند.

**آشنایی با API ها**

**1. تعریف:** یک API به عنوان مجموعه ای از قوانین عمل می کند که به یک برنامه نرم افزاری اجازه می دهد تا با نرم افزار دیگری تعامل داشته باشد. این روشی را برای توسعه دهندگان فراهم می کند تا به ویژگی ها یا داده های خاصی از یک سرویس، کتابخانه یا سیستم عامل دسترسی داشته باشند، بدون اینکه نیازی به درک عملکرد داخلی آن سیستم داشته باشند.

**2. انواع API:**

**Web API ها (HTTP/RESTful):** این API ها از پروتکل HTTP برای فعال کردن ارتباط از طریق وب استفاده می کنند. API های RESTful یک نوع رایج هستند که بر معماری بدون حالت و مقیاس پذیر تاکید دارند.

**API های کتابخانه:** زبان های برنامه نویسی اغلب با کتابخانه های استاندارد حاوی کد و توابع از پیش نوشته شده عرضه می شوند. API کتابخانه نحوه استفاده از این توابع را تعریف می کند.

**APIهای سیستم عامل:** این APIها به برنامه‌ها اجازه می‌دهند تا با سیستم عامل زیربنایی تعامل داشته باشند و به خدماتی مانند سیستم‌های فایل یا ارتباطات شبکه دسترسی پیدا کنند.

**3. اجزای یک API:**

**نقاط پایانی:** URL های خاص یا URI هایی که یک API برای تعامل با عملکردهای مختلف در معرض نمایش قرار می دهد.

**متد‌ها:** اقداماتی که معمولاً با استفاده از روش‌های HTTP مانند GET، POST، PUT و DELETE قابل انجام هستند.

**درخواست و پاسخ:** ارتباط بین برنامه‌ها شامل ارسال یک درخواست و دریافت پاسخ، معمولاً با فرمت JSON یا XML است.

**API ها چگونه کار می کنند**

1. **چرخه درخواست-پاسخ:** یک کلاینت درخواستی را به API ارسال می کند و عمل مورد نظر را مشخص می کند. API درخواست را پردازش می کند و اقدامات لازم را انجام می دهد. پاسخی به مشتری ارسال می‌شود که حاوی اطلاعات درخواستی است یا تکمیل عمل را تأیید می‌کند.

2. **احراز هویت و مجوز:** APIها اغلب به احراز هویت نیاز دارند تا اطمینان حاصل شود که فقط کاربران یا برنامه های مجاز می توانند به منابع خاصی دسترسی داشته باشند. مکانیسم های مجوز کنترل می کنند که کاربر یا برنامه چه اقداماتی می تواند انجام دهد. بهترین روش های طراحی API 1. سازگاری: در نامگذاری قراردادها، ساختارهای نقطه پایانی و فرمت های داده در سرتاسر API سازگاری داشته باشید. 2. نسخه سازی: برای اطمینان از اینکه تغییرات در API برنامه‌های موجود را که به نسخه‌های قدیمی‌تر تکیه می‌کنند، خراب نمی‌کند، نسخه‌سازی را پیاده‌سازی کنید.

3. **مستندات:** برای راهنمایی توسعه دهندگان در استفاده موثر از API، مستندات جامع و واضح ارائه کنید.

4. **امنیت:** از روش‌های ایمن، از جمله رمزگذاری (HTTPS)، مکانیسم‌های احراز هویت مناسب، و محدود کردن نرخ برای جلوگیری از سوء استفاده استفاده کنید.

**نمونه های دنیای واقعی**

1. Twitter API: به توسعه دهندگان اجازه می دهد تا به عملکردهای توییتر مانند ارسال توییت یا بازیابی اطلاعات کاربر دسترسی داشته باشند.

2. Google Maps API: ادغام نقشه های گوگل در برنامه ها را فعال می کند و قابلیت هایی مانند موقعیت جغرافیایی و برنامه ریزی مسیر را ارائه می دهد.

**نتیجه گیری**

در چشم انداز همیشه در حال تکامل توسعه نرم افزار، API ها نقشی محوری در تقویت قابلیت همکاری و همکاری بین سیستم های مختلف ایفا می کنند. طراحی و اجرای آنها به طور قابل توجهی بر قابلیت استفاده و موفقیت برنامه هایی که از آنها استفاده می کنند تأثیر می گذارد. همانطور که تکنولوژی به پیشرفت خود ادامه می دهد، APIها به عنوان یک بلوک اساسی برای ایجاد راه حل های نرم افزاری قوی و به هم مرتبط خواهند بود.

bard1

**رمزگشایی از رابط های برنامه نویسی برنامه (API)**

در دنیای توسعه نرم افزار، API ها (رابط برنامه نویسی برنامه) همه جا حاضر شده اند و به عنوان ستون فقرات برنامه های کاربردی مدرن عمل می کنند و امکان ارتباط یکپارچه بین اجزای مختلف نرم افزار را فراهم می کنند. در حالی که اصطلاح "API" ممکن است دلهره آور به نظر برسد، درک مفهوم اصلی و کاربردهای مختلف آن می تواند اهمیت و پذیرش گسترده آن را روشن کند.

**ماهیت API ها:** یک API به عنوان یک واسطه عمل می کند و ارتباط بین دو یا چند برنامه را تسهیل می کند. مجموعه ای از قوانین و مشخصاتی را تعریف می کند که بر نحوه تبادل داده ها و عملکردهای این برنامه ها نظارت می کند. درست مانند یک رابط کاربر را به یک کامپیوتر متصل می کند، یک API برنامه های نرم افزاری را به هم متصل می کند و به آنها اجازه می دهد تا اطلاعات را به اشتراک بگذارند و وظایف خاصی را بدون ادغام مستقیم انجام دهند.

**ساختار و اجزای یک API:** یک API معمولاً شامل سه جزء کلیدی است: درخواست‌ها، نقاط پایانی و پاسخ‌ها.

1. **درخواست ها:** اینها اساساً دستورالعمل هایی هستند که توسط یک برنامه به برنامه دیگر ارسال می شوند و داده ها یا اقدامات خاصی را درخواست می کنند.
2. **نقاط پایانی:** اینها مکان ها یا URL های تعیین شده ای هستند که در آن درخواست ها ارسال و پاسخ ها دریافت می شود. هر نقطه پایانی هدف خاصی را انجام می دهد که مربوط به یک عملکرد یا مجموعه داده خاص است.
3. **پاسخ ها:** این ها خروجی هایی هستند که توسط API در پاسخ به درخواست ها تولید می شوند. آنها ممکن است حاوی داده های درخواستی، پیام های خطا یا به روز رسانی وضعیت باشند.

**انواع API:** طبقه بندی های مختلفی از API ها بر اساس عملکرد و روش های دسترسی آنها وجود دارد. در اینجا چند نوع برجسته آورده شده است:

1. **APIهای عمومی:** اینها باز و آزادانه در دسترس هستند و به هر کسی اجازه می دهند با خدمات خود یکپارچه شوند. مثال‌ها عبارتند از Google Maps API، Facebook Graph API و Twitter API.
2. **API های خصوصی:** این ها API های اختصاصی هستند که به افراد یا سازمان های خاص محدود می شوند. آنها اغلب برای کنترل سیستم های داخلی یا ارائه عملکردهای سفارشی به کاربران مجاز استفاده می شوند.
3. **API های شریک:** این ها بین شرکت ها یا سازمان ها برای تسهیل همکاری و به اشتراک گذاری داده ها ایجاد می شوند. آنها معمولاً توسط توافق نامه ها و اقدامات امنیتی خاص اداره می شوند.

**کاربردهای رایج API ها:** API ها روش تعامل نرم افزار و قدرت بسیاری از برنامه ها را متحول کرده اند. در اینجا چند نمونه رایج آورده شده است: خدمات موقعیت مکانی: APIهایی مانند Google Maps و OpenStreetMap به برنامه‌ها اجازه می‌دهند تا نقشه‌ها، مسیرهای رانندگی و سرویس‌های مبتنی بر مکان را ادغام کنند.

1. **یکپارچه سازی رسانه های اجتماعی:** API هایی مانند Facebook Graph API و Twitter API برنامه ها را قادر می سازند تا داده های کاربر را به اشتراک بگذارند، پروفایل ها را متصل کنند و فیدهای رسانه های اجتماعی را نمایش دهند.
2. **پردازش پرداخت:** API هایی مانند PayPal و Stripe پرداخت ها و تراکنش های آنلاین امن بین برنامه ها و درگاه های پرداخت را تسهیل می کنند.
3. **تجسم داده ها:** API هایی مانند Chart.js و Highcharts ابزارهایی را برای ایجاد نمودارها و نمودارهای تعاملی مستقیماً در برنامه ها ارائه می دهند.

**مزایای API ها:** پذیرش APIها مزایای متعددی را برای اکوسیستم توسعه نرم افزار به ارمغان آورده است:

1. **قابلیت استفاده مجدد و کارایی:** APIها به توسعه دهندگان این امکان را می دهند تا از عملکرد و داده های موجود بدون تکرار تلاش های خود استفاده کنند. این کار استفاده مجدد از کد را ترویج می کند و کارایی توسعه را بهبود می بخشد.
2. **قابلیت همکاری:** APIها موانع بین سیستم‌های نرم‌افزاری مختلف را از بین می‌برند و ارتباطات و تبادل داده‌ها را یکپارچه می‌سازند. این امر قابلیت همکاری و نوآوری را تقویت می کند.
3. **مقیاس‌پذیری و چابکی:** APIها می‌توانند به راحتی نیازهای رو به رشد برنامه‌ها را برآورده کنند و به آن‌ها اجازه می‌دهند تا مقیاس‌پذیری و سازگاری با نیازهای در حال تغییر را داشته باشند.
4. **بهره وری توسعه دهنده:** APIها پیچیدگی توسعه را کاهش می دهند، اجزای از پیش ساخته شده و رابط های استاندارد شده را ارائه می دهند. این کار بهره وری توسعه دهندگان را افزایش می دهد و زمان ورود به بازار را کاهش می دهد.

**نتیجه گیری:** در نتیجه، APIها به ابزاری ضروری در چشم انداز نرم افزاری مدرن تبدیل شده اند که امکان ارتباط یکپارچه، تبادل داده و همکاری بین برنامه ها را فراهم می کند. پذیرش گسترده آنها شیوه تعامل ما با دنیای دیجیتال را متحول کرده است و آن را به هم پیوسته تر، پویاتر و پاسخگوتر به نیازهای ما کرده است. همانطور که صنعت نرم افزار همچنان به تکامل خود ادامه می دهد، بدون شک API ها نقش برجسته تری در شکل دادن به آینده فناوری و نحوه استفاده ما از آن خواهند داشت.

bard2

**رمزگشایی API ها: راهنمای جامع برای رابط های برنامه نویسی کاربردی**

در حوزه توسعه نرم‌افزار، APIها (واسط‌های برنامه‌نویسی کاربردی) به عنوان واسطه‌های همه‌جا حضور دارند و ارتباطات یکپارچه و تبادل داده بین برنامه‌ها را تسهیل می‌کنند. این ابزارهای بی ادعا و در عین حال قدرتمند شیوه تعامل ما با فناوری را متحول کرده است و توسعه دهندگان را قادر می سازد تا برنامه های به هم پیوسته و غنی از ویژگی ها را بسازند.

**درک ماهیت API ها**

در هسته خود، APIها به عنوان واسطه عمل می کنند و شکاف بین برنامه ها را پر می کنند. آنها مجموعه ای از قوانین و مشخصات تعریف شده را ارائه می کنند که نحوه تعامل یک برنامه با برنامه دیگر را کنترل می کند. این تعامل معمولاً شامل ارسال درخواست‌ها و دریافت پاسخ است که به برنامه‌ها اجازه می‌دهد تا اقدامات خاصی را انجام دهند یا داده‌ها را بازیابی کنند.

**آناتومی یک API**

یک API از چندین جزء کلیدی تشکیل شده است:

**نقاط پایانی:** مکان‌های تعیین‌شده‌ای هستند که در آن درخواست‌ها ارسال و پاسخ‌ها دریافت می‌شوند. آنها نشان دهنده نقاط ورود به عملکرد یک API هستند.

**متد های درخواست:** این افعال استاندارد شده ای هستند که برای تعیین نوع عملیات مورد درخواست استفاده می شوند. متدهای رایج عبارتند از GET، POST، PUT و DELETE.

**فرمت داده:** APIها از فرمت های داده خاصی برای ساختار دادن به اطلاعات مبادله شده بین برنامه ها استفاده می کنند. فرمت های رایج عبارتند از JSON (جاوا اسکریپت نشانه گذاری شی) و XML (زبان نشانه گذاری توسعه پذیر).

**احراز هویت:** API ها اغلب از مکانیسم های احراز هویت برای تأیید هویت برنامه درخواست کننده و اطمینان از دسترسی ایمن به منابع محافظت شده استفاده می کنند.

**مستندسازی:** مستندات واضح و جامع برای توسعه‌دهندگان برای درک ویژگی‌های API، دستورالعمل‌های استفاده و روش‌های رسیدگی به خطا بسیار مهم است.

**انواع API ها**

دنیای APIها متنوع است و انواع مختلفی را در بر می گیرد که اهداف خاصی را تامین می کنند. در اینجا چند دسته بندی رایج وجود دارد:

**APIهای عمومی:** این APIها به طور آشکار در دسترس عموم هستند و دسترسی رایگان به عملکرد خود را فراهم می کنند. به عنوان مثال می توان به API های رسانه های اجتماعی، API های آب و هوا و API های نقشه برداری اشاره کرد.

**API های شریک:** این API ها در دسترس شرکا یا مشتریان منتخب قرار می گیرند و اغلب به توافق نامه مجوز یا اشتراک نیاز دارند. آنها دسترسی به داده های اختصاصی یا ویژگی های منحصر به فرد شرکا را فراهم می کنند.

**APIهای خصوصی:** این APIها کاملاً داخلی یک سازمان هستند و برای ارتباطات و تبادل داده بین برنامه های داخلی آن استفاده می شوند.

**API های وب:** این API ها از طریق فناوری های وب مانند HTTP قابل دسترسی هستند و به برنامه ها اجازه می دهند از طریق درخواست ها و پاسخ های وب با آنها تعامل داشته باشند.

**مزایای API ها**

شیوع API ها از مزایای متعدد آنها ناشی می شود:

**قابلیت استفاده مجدد:** APIها به توسعه دهندگان این امکان را می دهند که از کد و عملکردی که توسط دیگران ایجاد شده است استفاده مجدد کنند و در زمان و تلاش صرفه جویی کنند.

**یکپارچه سازی:** API ها یکپارچه سازی یکپارچه بین برنامه ها و پلتفرم های مختلف را تسهیل می کنند، سیلوهای داده را تجزیه می کنند و عملکرد کلی را بهبود می بخشند.

**نوآوری:** APIها نوآوری را با امکان توسعه دهندگان برای ایجاد برنامه های کاربردی و تجربیات جدید تقویت می کنند.

**مقیاس‌پذیری:** APIها می‌توانند حجم زیادی از درخواست‌ها را به طور کارآمد انجام دهند، و آنها را برای برنامه‌هایی با تقاضای ترافیک بالا مناسب می‌سازد.

**دسترس‌پذیری:** APIها روشی استاندارد برای ارتباط برنامه‌ها ارائه می‌کنند که سازگاری و قابلیت همکاری را ارتقا می‌دهند.

**کاربردهای API ها**

کاربردهای APIها گسترده و دائماً در حال تکامل هستند و طیف وسیعی از صنایع و موارد استفاده را در بر می گیرند. در اینجا چند نمونه هستند:

**تجارت الکترونیک:** APIها تجربه خرید آنلاین را تقویت می کنند و به کاربران امکان می دهند محصولات را مرور کنند، خرید کنند و حساب های خود را مدیریت کنند.

**خدمات مالی:** APIها بانک ها و مؤسسات مالی را قادر می سازند تا دسترسی ایمن و راحت به اطلاعات حساب، تراکنش ها و پردازش پرداخت را فراهم کنند. رسانه‌های اجتماعی: APIها ویژگی‌های رسانه‌های اجتماعی مختلف مانند اشتراک‌گذاری محتوا، ارسال به‌روزرسانی‌ها و ادغام با برنامه‌های خارجی را تقویت می‌کنند.

**نقشه‌برداری و ناوبری:** APIها دسترسی به نقشه‌ها، مسیرها و خدمات مبتنی بر مکان را فراهم می‌کنند و برنامه‌های ناوبری و مکان‌یابی را فعال می‌کنند.

**اینترنت اشیا و دستگاه های متصل:** API ها دستگاه های اینترنت اشیا را به هم متصل می کنند و آنها را قادر می سازند تا داده ها را مبادله کنند و با سرویس های مبتنی بر ابر ارتباط برقرار کنند.

**نتیجه گیری**

APIها به ابزارهای ضروری در چشم انداز نرم افزار مدرن تبدیل شده اند و نحوه تعامل و اشتراک گذاری داده ها را متحول می کنند. تطبیق پذیری و توانایی آنها برای اتصال سیستم های متفاوت باعث نوآوری و دگرگونی صنایع در سراسر جهان شده است. همانطور که تکنولوژی به تکامل خود ادامه می دهد، بدون شک API ها نقش بیشتری در شکل دادن به آینده توسعه نرم افزار و تجربیات کاربر خواهند داشت.

bard3

**رمزگشایی API ها: راهنمای جامع برای رابط های برنامه نویسی کاربردی**

در قلمرو همیشه در حال تکامل توسعه نرم افزار، نام اختصاری "API" در همه جا رایج شده است، که اغلب در بحث های بین توسعه دهندگان، کارآفرینان و حتی علاقه مندان به فناوری معمولی مطرح می شود. اما API دقیقاً چیست و چرا در چشم انداز فناوری امروزی چنین برجسته است؟

**API: رونمایی از رابط**

API مخفف Application Programming Interface، یک واسطه نرم افزاری است که به دو برنامه اجازه می دهد تا با یکدیگر ارتباط برقرار کنند. به عنوان یک پیام رسان عمل می کند، درخواست ها و پاسخ ها را بین برنامه ها ارسال می کند و آنها را قادر می سازد تا داده ها و عملکردها را به طور یکپارچه مبادله کنند. سناریویی را در نظر بگیرید که در آن می خواهید پرواز آنلاین رزرو کنید. وب سایت شرکت هواپیمایی به عنوان یک برنامه کاربردی عمل می کند، در حالی که مرورگر وب شما به عنوان برنامه دیگر عمل می کند. برای انجام فرآیند رزرو، مرورگر شما درخواست هایی را به وب سایت شرکت هواپیمایی ارسال می کند، مانند "چه پروازهایی در این تاریخ موجود است؟" یا "هزینه رزرو بلیط درجه یک چقدر است؟" وب سایت شرکت هواپیمایی نیز به نوبه خود با اطلاعات درخواستی پاسخ می دهد و مرورگر شما را قادر می سازد گزینه های پرواز را نمایش دهد و فرآیند رزرو را تکمیل کند. در اصل، API به عنوان پل ارتباطی بین مرورگر شما و وب‌سایت شرکت هواپیمایی عمل می‌کند و به آن‌ها اجازه می‌دهد با یکدیگر ارتباط برقرار کنند و تراکنش رزرو را تسهیل کنند.

**آناتومی یک API**

یک API شامل چندین مؤلفه است که برای ایجاد ارتباط بین برنامه ها با هم کار می کنند:

1. **نقاط پایانی:** اینها نشانی‌های اینترنتی یا آدرس‌هایی هستند که برنامه‌ها از آنها برای ارسال و دریافت درخواست‌ها و پاسخ‌ها استفاده می‌کنند. هر نقطه پایانی عملکرد خاصی مانند جستجوی داده ها، ایجاد رکوردهای جدید یا به روز رسانی اطلاعات موجود را انجام می دهد.
2. **مستندات:** یک مستند جامع API اطلاعات دقیقی درباره نقاط پایانی API، استفاده از آنها، فرمت‌های داده‌ای که می‌پذیرند و برمی‌گردانند و هر مکانیزم رسیدگی به خطا را ارائه می‌دهد.
3. **فرمت‌های داده:** APIها از فرمت‌های داده استاندارد شده، مانند JSON (JavaScript Object Notation) یا XML (Extensible Markup Language) برای ساختار و تبادل اطلاعات بین برنامه‌ها استفاده می‌کنند.
4. **امنیت:** APIها اغلب از اقدامات امنیتی مختلفی مانند احراز هویت و مجوز برای محافظت از داده های حساس و جلوگیری از دسترسی غیرمجاز استفاده می کنند.

**قدرت API ها: طیفی از برنامه ها**

API ها نقشی محوری در توسعه نرم افزار مدرن ایفا می کنند و امکان ادغام و همکاری یکپارچه بین برنامه ها و سرویس های مختلف را فراهم می کنند. کاربردهای آنها در دامنه های مختلف گسترش می یابد:

1. **خدمات وب:** APIها ستون فقرات خدمات وب را تشکیل می دهند و ارتباط بین برنامه های کاربردی وب و ارائه دهندگان خدمات وب را امکان پذیر می سازند.
2. **برنامه های کاربردی تلفن همراه:** API ها به برنامه های تلفن همراه اجازه دسترسی و ادغام داده ها از منابع خارجی را می دهند و تجربه کاربری غنی تری را ارائه می دهند.
3. **اینترنت اشیا (IoT): API** ها تبادل داده بین دستگاه های اینترنت اشیا و سرویس های مبتنی بر ابر را تسهیل می کنند و امکان نظارت و کنترل از راه دور را فراهم می کنند.
4. **یکپارچه‌سازی رسانه‌های اجتماعی:** APIها پلتفرم‌های رسانه‌های اجتماعی را قادر می‌سازند تا با سایر برنامه‌ها یکپارچه شوند و به کاربران اجازه می‌دهند محتوا را به اشتراک بگذارند و بین پلتفرم‌ها ارتباط برقرار کنند.

**API ها: نیروی محرکه نوآوری** API ها روش توسعه و مصرف نرم افزار را متحول کرده اند. آنها دسترسی به داده ها و عملکرد را دموکراتیک کرده اند و توسعه دهندگان را قادر می سازند تا برنامه ها و خدمات نوآورانه ای را بدون اختراع مجدد چرخ بسازند. با استفاده از API های موجود، توسعه دهندگان می توانند روی ایجاد ارزش های پیشنهادی منحصر به فرد و تجربیات کاربر تمرکز کنند. APIها همچنین همکاری و نوآوری را در بین صنایع تقویت کرده اند، سیلوها را شکسته و امکان یکپارچه سازی یکپارچه خدمات و داده ها را فراهم کرده اند. این امر سرعت نوآوری را تسریع کرده و کسب و کارها را قادر می سازد تا به سرعت با روندهای در حال تحول بازار سازگار شوند.

**نتیجه گیری:** API ها آینده نرم افزار را شکل می دهند APIها به ابزارهای ضروری در چشم انداز توسعه نرم افزار تبدیل شده اند و نحوه ارتباط برنامه ها و به اشتراک گذاری داده ها را تغییر می دهند. تطبیق پذیری، تطبیق پذیری و توانایی آنها برای تقویت همکاری، آنها را به سنگ بنای توسعه نرم افزار مدرن تبدیل کرده است. همانطور که تکنولوژی به تکامل خود ادامه می دهد، بدون شک API ها نقش مهمی در شکل دادن به آینده نرم افزار و نحوه تعامل ما با فناوری خواهند داشت.

poe

درک رابط های برنامه نویسی کاربردی (API):

معرفی: در دنیای توسعه نرم افزار، رابط های برنامه نویسی کاربردی (API) نقشی اساسی در تسهیل ارتباطات و تعامل بین سیستم های نرم افزاری مختلف ایفا می کنند. APIها به عنوان بلوک های سازنده ای عمل می کنند که توسعه دهندگان را قادر می سازد تا با استفاده از قابلیت های خدمات نرم افزاری، کتابخانه ها یا پلتفرم های موجود، برنامه های کاربردی قوی ایجاد کنند. در این مقاله مروری کامل و جامع از APIها، هدف، انواع و نحوه استفاده از آنها در برنامه نویسی ارائه می شود.

فهرست مطالب:

API چیست؟

هدف از API ها

انواع API ها

3.1. APIهای وب (HTTP/RESTful API)

3.2. کتابخانه یا چارچوب API

3.3. API های سیستم عامل

3.4. APIهای پایگاه داده

3.5. API های SOAP

3.6. سایر API های تخصصی

API ها چگونه کار می کنند

اصول طراحی API

پروتکل های ارتباطی مشترک API 6.1. HTTP (پروتکل انتقال ابرمتن) 6.2. REST (انتقال دولتی نمایندگی) 6.3. JSON (نشان‌گذاری شی جاوا اسکریپت) 6.4. XML (زبان نشانه گذاری توسعه یافته) 6.5. SOAP (پروتکل دسترسی به اشیاء ساده) احراز هویت و امنیت API 7.1. کلیدهای API 7.2. OAuth (مجوز باز) 7.3. JWT (JSON Web Tokens) 7.4. محدود کردن نرخ اسناد API تست API و اشکال زدایی بهترین روش ها برای مصرف API ها ساخت API ها نسخه API و سازگاری به عقب مدیریت چرخه حیات API نتیجه API چیست؟ رابط برنامه نویسی برنامه (Application Programming Interface) یا API مجموعه ای از قوانین، پروتکل ها و ابزارهایی است که به نرم افزارهای مختلف اجازه می دهد تا با یکدیگر ارتباط برقرار کرده و با یکدیگر تعامل داشته باشند. قراردادی را بین ارائه‌دهنده یک سرویس (ارائه‌دهنده API) و مصرف‌کننده (کاربر API) تعریف می‌کند و نحوه تعامل آنها با یکدیگر را به صورت برنامه‌ریزی مشخص می‌کند. هدف API ها: APIها اهداف مختلفی را انجام می دهند، از جمله: توسعه دهندگان را قادر می سازد تا از عملکردها و خدمات موجود استفاده کنند و در زمان و تلاش در اختراع مجدد چرخ صرفه جویی کنند. ارتقاء مدولار بودن و قابلیت استفاده مجدد با کپسوله کردن عملکردهای پیچیده در یک رابط کاملاً تعریف شده. تسهیل اتصال بین سیستم های نرم افزاری مختلف، به آنها اجازه می دهد تا داده ها را مبادله کنند و به طور یکپارچه با هم کار کنند. ساده سازی توسعه نرم افزار با انتزاع کردن جزئیات پیچیده پیاده سازی و ارائه انتزاعات سطح بالا. تقویت همکاری با اجازه دادن به توسعه دهندگان برای ساخت برنامه های کاربردی بر روی پلتفرم های موجود و گسترش قابلیت های خود. انواع API: بسته به کاربرد مورد نظرشان، چندین نوع API وجود دارد: 3.1. APIهای وب (HTTP/RESTful API): این APIها برای برقراری ارتباط بین برنامه های کاربردی مبتنی بر وب طراحی شده اند. آنها از پروتکل های استاندارد وب مانند HTTP و REST برای دسترسی به منابع و انجام عملیات بر روی آنها استفاده می کنند. 3.2. کتابخانه یا Framework API: این APIها در کتابخانه‌ها یا چارچوب‌های نرم‌افزاری قرار می‌گیرند و توابع، کلاس‌ها و ماژول‌های از پیش تعریف‌شده‌ای را ارائه می‌کنند که توسعه‌دهندگان می‌توانند از آنها برای ساخت برنامه‌ها استفاده کنند. 3.3. APIهای سیستم عامل: سیستم‌های عامل APIهایی را در معرض نمایش می‌گذارند که به توسعه‌دهندگان اجازه می‌دهد با منابع سیستمی مختلف مانند سیستم‌های فایل، رابط‌های شبکه یا دستگاه‌های سخت‌افزار تعامل داشته باشند. 3.4. APIهای پایگاه داده: APIهای پایگاه داده ابزاری برای تعامل با پایگاه‌های داده فراهم می‌کنند و به توسعه‌دهندگان اجازه می‌دهند تا داده‌ها را به‌صورت برنامه‌نویسی جستجو، درج، به‌روزرسانی و حذف کنند. 3.5. API های SOAP: API های پروتکل دسترسی ساده به اشیا (SOAP) از پروتکل های پیام رسانی مبتنی بر XML برای فعال کردن ارتباط بین برنامه ها از طریق شبکه استفاده می کنند. 3.6. سایر API های تخصصی: API های تخصصی متعددی برای اهداف خاص در دسترس هستند، مانند API های رسانه های اجتماعی، API های درگاه پرداخت، API های موقعیت جغرافیایی و غیره. API ها چگونه کار می کنند: API ها بر اساس یک مدل درخواست-پاسخ کار می کنند. ارائه‌دهنده API مجموعه‌ای از نقاط پایانی (URL) را نشان می‌دهد که هر کدام یک عملکرد یا منبع خاصی را نشان می‌دهند. مصرف‌کننده API (توسعه‌دهنده) درخواست‌هایی را به این نقاط پایانی ارسال می‌کند و اقدام مورد نظر را مشخص می‌کند و داده‌های مورد نیاز را ارائه می‌کند. ارائه‌دهنده API درخواست را پردازش می‌کند و پاسخی را معمولاً در قالب داده‌ای خاص مانند JSON یا XML ارسال می‌کند. این پاسخ حاوی داده های درخواستی است یا نشان دهنده موفقیت/شکست عملیات است. اصول طراحی API: APIهای خوب طراحی شده از اصول خاصی پیروی می کنند که قابلیت استفاده و نگهداری آنها را افزایش می دهد: قراردادهای نامگذاری واضح و شهودی برای نقاط پایانی، روش ها و پارامترها. رفتار سازگار و قابل پیش بینی در نقاط پایانی مختلف. واسط های مستند و خود توضیحی. مدیریت صحیح خطا و کدهای وضعیت. مکانیزم‌های نسخه‌سازی برای مدیریت سازگاری به عقب پروتکل های ارتباطی مشترک API: برای تسهیل ارتباطات و تبادل داده ها، API ها بر پروتکل ها و فرمت های مختلفی تکیه می کنند: 6.1. HTTP (پروتکل انتقال ابرمتن): پایه و اساس ارتباطات وب است که به مشتریان و سرورها امکان تبادل داده را می دهد. 6.2. REST (انتقال حالت نمایندگی): مجموعه‌ای از اصول و محدودیت‌های معماری که بهترین روش‌ها را برای ساخت APIهای وب مقیاس‌پذیر و بدون حالت تعریف می‌کنند. 6.3. JSON (JavaScript Object Notation): یک فرمت تبادل داده سبک است که معمولاً برای ساختاردهی و انتقال داده ها بین APIها و کلاینت ها استفاده می شود. 6.4. XML (EXtensible Markup Language): فرمت دیگری برای ساختاردهی و انتقال داده ها که اغلب در سیستم های قدیمی یا صنایع خاص استفاده می شود. 6.5. SOAP (Simple Object Access Protocol): پروتکلی برای تبادل اطلاعات ساختاریافته در سرویس های وب با استفاده از XML. احراز هویت و امنیت API: API ها اغلب به احراز هویت و اقدامات امنیتی برای اطمینان از دسترسی مجاز و محافظت از داده های حساس نیاز دارند. برخی از روش های رایج عبارتند از: 7.1. کلیدهای API: یک شناسه منحصر به فرد ارائه شده به مصرف کنندگان API برای احراز هویت درخواست های آنها. 7.2. OAuth (مجوز باز): پروتکلی که مجوز ایمن و تفویض اختیار دسترسی بین سیستم های مختلف را امکان پذیر می کند. 7.3. JWT (JSON Web Tokens): یک روش فشرده و مستقل برای انتقال امن اطلاعات بین طرفین به عنوان یک شی JSON. 7.4. محدود کردن نرخ: برای جلوگیری از سوء استفاده و اطمینان از استفاده منصفانه، محدودیت‌هایی بر تعداد درخواست‌هایی که مشتری می‌تواند در یک بازه زمانی معین انجام دهد اعمال می‌کند. اسناد API: مستندات جامع و ساختارمند API برای توسعه دهندگان برای درک قابلیت ها، الزامات و استفاده از یک API بسیار مهم است. معمولاً شامل جزئیات مربوط به نقاط پایانی، فرمت‌های درخواست/پاسخ، روش‌های احراز هویت، رسیدگی به خطا و نمونه‌های کد است. تست و رفع اشکال API: آزمایش و اشکال زدایی کامل APIها برای اطمینان از عملکرد، قابلیت اطمینان و عملکرد آنها ضروری است. مجموعه‌های تست، تست‌های واحد، تست‌های یکپارچه‌سازی و تست‌های بار را می‌توان برای اعتبارسنجی رفتار API و شناسایی و حل مشکلات استفاده کرد. بهترین روش ها برای مصرف API ها: هنگام مصرف APIها، توسعه دهندگان باید از بهترین شیوه ها پیروی کنند، از جمله: مدیریت صحیح خطاها و استثناهای برگردانده شده توسط API. پیاده سازی مکانیسم های مناسب ذخیره سازی برای بهبود عملکرد. بهینه سازی تماس های API با کاهش درخواست های غیر ضروری و اندازه بار. اجرای راهبردهای تکرار و رسیدگی به خطا برای مسائل مربوط به شبکه. ذخیره و مدیریت ایمن اعتبارنامه های API. ساخت API: برای ساخت API، توسعه دهندگان یک سری مراحل را دنبال می کنند، از جمله: تعریف هدف، عملکرد و مخاطبان هدف API. طراحی نقاط پایانی API، ساختارهای داده و فرمت‌های درخواست/پاسخ. پیاده سازی منطق تجاری API و ادغام آن با منابع داده یا خدمات مرتبط. آزمایش و اعتبارسنجی عملکرد و عملکرد API. مستندسازی API به طور کامل برای کمک به توسعه دهندگانی که آن را مصرف می کنند. نسخه API و سازگاری به عقب: همانطور که API ها در طول زمان تکامل می یابند، مدیریت نسخه سازی و سازگاری به عقب ضروری می شود. نسخه‌سازی به توسعه‌دهندگان اجازه می‌دهد تا تغییرات را ارائه کنند و در عین حال اطمینان حاصل شود که مشتریان موجود به کار خود ادامه می‌دهند. برای مدیریت مؤثر نسخه‌های API می‌توان از تکنیک‌هایی مانند نسخه‌سازی معنایی، نسخه‌سازی API در URLها یا استفاده از هدرهای سفارشی استفاده کرد. مدیریت چرخه عمر API: مدیریت چرخه عمر API شامل مراحل مختلفی از جمله برنامه ریزی، طراحی، توسعه، آزمایش، استقرار، نگهداری و بازنشستگی است. مدیریت صحیح، طول عمر، عملکرد، امنیت و بهبود مستمر API را تضمین می کند. نتیجه: APIها برای توسعه نرم افزار مدرن اساسی هستند و امکان یکپارچه سازی یکپارچه، همکاری و ایجاد برنامه های کاربردی نوآورانه را فراهم می کنند. درک هدف، انواع، اصول طراحی، ملاحظات امنیتی و بهترین شیوه‌های مرتبط با APIها، به توسعه‌دهندگان این امکان را می‌دهد تا از پتانسیل کامل این رابط‌ها استفاده کنند و سیستم‌های قوی، مقیاس‌پذیر و به هم پیوسته بسازند.