

Let's dive into optimizing API performance. First up, pagination. When you've got tons of data to send, bring it into smaller chunks or pages. This way, your service dates quick, and users don't have the way forever. Next, asynchronous logging. Instead of logging that writes to the disk every time. Gather log data in memory and write it to the disk now and then. This cuts down on waiting time for disk operations. Third, caching. Use something like redds to cache data you use a lot. Redds let you get to your data super fast. First, check the cache and only ask the database if the cache doesn't have what you need. Fourth, payload compression. Shring the size of the data you're sending and receiving with tools like G-ZIP. Smaller data equals faster uploads and downloads. Finally, connection pool. Avoid the overhead of opening and closing database connections all the time. A connection pool keeps these connections ready to go, making working with your databases much faster. Master system design get 158 page system design ebook for free.