ByteByteGo
592,569 followers
41m · 🌐

How do we design effective and safe APIs?

The diagram below shows typical API designs with a shopping cart example.

Note that API design is not just URL path design. Most of the time, we need to choose the proper resource names, identifiers, and path patterns. It is equally important to design proper HTTP header fields or to design effective rate-limiting rules within the API gateway.

Over to you: What are the most interesting APIs you've designed?

--

Subscribe to our weekly newsletter to get a Free System Design PDF (158 pages): https://bit.ly/bbg-social

#systemdesign #coding #interviewtips

.

Design Effective & Safe APIs

Design a Shopping Cart

	❌	✅
Use resource names (nouns)	GET /querycarts/123	GET /carts/123
Use plurals	GET /cart/123	GET /carts/123
Idempotency	POST /carts	POST /carts {requestId: 4321}
Use versioning	GET /carts/v1/123	GET /v1/carts/123
Query after soft deletion	GET /carts	GET /carts? includeDeleted=true
Pagination	GET /carts	GET /carts? pageSize=xx&pageToken=xx
Sorting	GET /items	GET /items? sort_by=time
Filtering	GET /items	GET /items? filter=color:red
Secure Access	X-API-KEY=xxx	X-API-KEY = xxx X-EXPIRY = xxx X-REQUEST-SIGNATURE = xxx → hmac(URL + QueryString + Expiry + Body)
Resource cross reference	GET /carts/123? item=321	GET /carts/123/items/321
Add an item to a cart	POST /carts/123? addItem=321	POST /carts/123/items:add { itemId: "items/321" }
Rate limit	No rate limit - DDos	Design rate limiting rules based on IP, user, action group etc

Alex Xu and 104 others   8 reposts