1 CheatSheet: System Design For Code Interview INTERVIEW

- PDF Link: cheatsheet-behavior-A4.pdf, Category: interview
- Blog URL: `https://cheatsheet.dennyzhang.com/cheatsheet-systemdesign-A4`
- Related posts: CheatSheet: Leetcode For Code Interview, #denny-cheatsheets

File me Issues or star this repo.

1.1 Process Of System Design

Num	Name	Summary
1	List use cases: List major and focus on some	Good sense. The questions you asked define your level
2	Understand scale: **Data** + **Traffic**	Estimate the volume of the problem space
3	Abstract design	
4	Bottlenecks: **Key challenges** + **Trade-Offs**	Valid reasoning is the key
5	Scale your design	

1.2 Engineering Of Well-Known Products

Name	Summary
Google	Link: Google Architecture
Facebook	Link: Facebook Live Streams
Twitter	Link: Twitter Image Service, YouTube: Timelines at Scale
Uber	Link: Lessons Learned From Scaling Uber
Tumblr	Link: Tumblr Architecture
StackOverflow	Link: Stack Overflow Architecture

1.3 Engineering Blogs/Websites

Name	Summary
Compnay Tech Blog	Website: Facebook Engineering, Website: Google Developers
Compnay Tech Blog	Medium: Netflix Blog, Medium: Airbnb Engineering & Data Science
Individual Tech Blog	Blog: All Things Distributed - Amazon CTO, Blog: highscalability
Website	Website: interviewing.io, Website: interviewbit.com
Website	Website: hiredintech - System Design, Github: system-design-primer
YouTube	YouTube Channel: Success in Tech, YouTube: Scalability Harvard Web Development
Reference	Link: Preparing for your Software Engineering Interview at Facebook
Reference	Link: The System Design Process
Cheatsheet	Code problems for #oodesign, CheatSheet: System Design For Code Interview
Cheatsheet	CheatSheet: Leetcode For Code Interview
Cheatsheet	CheatSheet: Behavior Questions For Coder Interview

1.4 Classic Design Problems - Design Business Features

Num	Name	Summary
1	Design: TinyURL - A URL Shorterner Service	
2	Design news feed	
3	Design web crawler	
4	Google autocomplete	
5	Design: A Parking Lot Service	
6	Design: An Elevator Service	
7	Design: A URL Redirecting Feature	
8	Design Google PageRank	

1.5 Classic Design Problems - Design A Complex Product

- Think through the complicated, high-scale systems that you already use every day

Num	Name	Summary
1	Design: Uber Backend	
2	Design K/V DB	
3	Design: Google Suggestion Service	
4	Design twitter	
5	Deisgn a payment processor	
6	Design a voice conference system	
7	Design google doc	
8	Design gmail	

1.6 Classic Design Problems - Design Technical Modules

Num	Name	Summary
1	Design a distributed UUID generator	
2	Design An API Rate Limiter	link, link
3	Design a circuit breaker	
4	Design a stack supporting push/pop/getmin/getmostfrequent	
5	Implement a timer	
6	Thread-safe Hashmap	
7	Top URL hits	
8	Unique url hits	
9	Delayed task scheduling	
10	Design a client-server API to build a rich document editor	

1.7 Classic Design Problems - Explain workflow: What happens when XXX?

Num	Name	Summary
1	When happens when I search in google?	
2	How loadbalancer works	
3	Explain three phase commit model	
4	Explain HTTP return code	301 vs 302, 401 vs 403, etc
5	Explain mysql DB replication model	
6	Explain Mysql replication mechanism	
7	Explain gossip protocol	
8	Explain CAP	

1.8 Classic Design Problems - Explain tools: how XXX supports XXX?

Num	Name	Summary
1	How JDK implement hashmap?	
2	Explain java garbage collection model	
3	Explain raft/etcd	

1.9 Classic Design Problems - Big Data

Num	Name	Summary
1	Reservoir sampling	
2	Frequency estimation	
3	Heavy hitters - Find top k frequent items in a data stream	
4	Membership query - whether an element exists in a data stream	bloom filter
5	Get median from an endless data stream	

1.10 Grow Design Expertise In Daily Work

Num	Summary
1	Think through the complicated, high-scale systems that you already use every day
2	Read engineering blogs of big companies
3	For common tools/frameworks, understand behind the hook
4	Try tools. What problem it solves, what its alternatives are, and what some common pros and cons may be.
5	Keep researching the alternatives to a technology. e.g, rabbitmq vs kafaka, memcached vs redis

1.11 More Resources

`https://github.com/binhnguyennus/awesome-scalability`