

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 Key Points Of System Design

Num	Name	Summary
1	The questions you asked define your level	Clarification; Painpoints; Key Points
2	What are the key trade-off? And your analysis	
3	Engage the discussion as a two-way conversation	

1.2 Resources

Name	Summary
Website	Github: system-design-primer , Website: interviewing.io , Website: interviewbit.com
Website	Website: hiredintech - System Design, YouTube: Success in Tech
Website	Link: Preparing for your Software Engineering Interview at Facebook
Cheatsheet	Code problems for #oodeesign , CheatSheet: System Design For Code Interview
Cheatsheet	CheatSheet: Leetcode For Code Interview
Cheatsheet	CheatSheet: Behavior Questions For Coder Interview

1.3 Classic Design Problems - Design Business Features

Num	Name	Summary
1	Design: TinyURL - A URL Shortener 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	

1.4 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	Design a payment processor	
6	Design a voice conference system	
7	Design google doc	
8	Design gmail	

1.5 Classic Design Problems - Design Technical Modules

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

1.6 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	

1.7 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	

1.8 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	

1.9 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

1.10 More Resources

License: Code is licensed under MIT License.

<https://github.com/binhnguyennus/awesome-scalability>