

System Design Websites

GitHub

- [System Design Preparation](#)
 - How to prepare for and answer system design questions
 - Starting point
 - basics
 - How to answer in interviews
 - Steps how I approach the system design questions in interviews
 - Common Design questions
 - architecture
 - company engineering blog links
 - Low on time ?
- [How to prepare system design questions for an IT company.](#)
 - System Design Interview Tips
 - Basic Knowledge about System Design
 - Company Engineering Blogs
 - Products and Systems
 - Hot Questions and Reference
 - Good Books
 - Object Oriented Design
- [Tech Interview Handbook](#)
 - How to prepare for coding interviews
 - Interview Cheatsheet - Straight-to-the-point Do's and Don'ts
 - Algorithm tips and the best practice questions categorized by topic
 - "Front-end Job Interview Questions" answers
 - Interview formats of the top tech companies
 - Behavioral questions asked by the top tech companies
 - Good questions to ask your interviewers at the end of the interviews
 - Helpful resume tips to get your resume noticed and the Do's and Don'ts
- [System Design and Architecture](#)
 - 实践
 - 短网址系统
 - Uber
 - Facebook 储存大规模社交图谱
 - Apache Kafka
 - Facebook 图片储存系统
 - HTTP 协议向移动设备传输视频 (HTTP Live Streaming)
 - Lyft 的营销自动化平台 (Symphony)
 - 负载均衡器
 - i18n 工程方案
 - 区块链服务端架构

- 理论
 - Introduction to Architecture
 - How to scale a web service?
 - ACID vs BASE
 - Data Partition and Routing
 - Replica, Consistency, and CAP theorem
 - Load Balancer Types
 - Concurrency Model
 - Improving availability with failover
 - Bloom Filter
 - Skiplist
 - B tree vs. B+ tree
 - Intro to Relational Database
 - 4 Kinds of No-SQL
 - Key value cache
 - Stream and Batch Processing Frameworks
 - Cloud Design Patterns
 - Public API Choices
 - Lambda Architecture
 - iOS Architecture Patterns Revisited
 - What can we communicate in soft skills interview?
 - Experience Deep Dive
 - 3 Programming Paradigms
 - SOLID Design Principles
- [System Design Primer](#)
 - 系统设计入门
 - 性能与扩展性
 - 延迟与吞吐量
 - 可用性与一致性
 - 一致模式
 - 可用模式
 - 域名系统
 - CDN
 - 负载均衡器
 - 反向代理
 - 应用层
 - 数据库
 - 缓存
 - 异步
 - 通讯
 - 安全
 - 附录

Links

Here are some articles about system design related topics.

- [How to Rock a Systems Design Interview](#)
 - [System Interview](#)
 - [Scalability for Dummies](#)
 - [Scalable Web Architecture and Distributed Systems](#)
 - [Numbers Everyone Should Know](#)
 - [Scalable System Design Patterns](#)
 - [Introduction to Architecting Systems for Scale](#)
 - [Transactions Across Datacenters](#)
 - [A Plain English Introduction to CAP Theorem](#)
 - [The CAP FAQ](#)
 - [Paxos Made Simple](#)
 - [Consistent Hashing](#)
 - [NOSQL Patterns](#)
 - [Scalability, Availability & Stability Patterns](#)
-

Hot Questions and Reference:

Design a CDN network

Reference:

- [Globally Distributed Content Delivery](#).

Design a Google document system

Reference:

- [google-mobwrite](#)
- [Differential Synchronization](#).

Design a random ID generation system

Reference:

- [Announcing Snowflake](#)
- [snowflake](#).

Design a key-value database

Reference:

- [Introduction to Redis](#).

Design the Facebook news feed function

Reference:

- [What are best practices for building something like a News Feed?](#)
- [What are the scaling issues to keep in mind while developing a social network feed?](#)
- [Activity Feeds Architecture](#)

Design the Facebook timeline function

Reference:

- [Building Timeline](#)
- [Facebook Timeline](#).

Design a function to return the top k requests during past time interval

Reference:

- [Efficient Computation of Frequent and Top-k Elements in Data Streams](#)
- [An Optimal Strategy for Monitoring Top-k Queries in Streaming Windows](#)

Design an online multiplayer card game

Reference:

- [How to Create an Asynchronous Multiplayer Game](#)
- [How to Create an Asynchronous Multiplayer Game Part 2: Saving the Game State to Online Database](#)
- [How to Create an Asynchronous Multiplayer Game Part 3: Loading Games from the Database](#)
- [How to Create an Asynchronous Multiplayer Game Part 4: Matchmaking](#)
- [Real Time Multiplayer in HTML5](#)

Design a graph search function

Reference:

- [Building out the infrastructure for Graph Search](#)
- [Indexing and ranking in Graph Search](#)
- [The natural language interface of Graph Search](#) and [Erlang at Facebook](#).

Design a picture sharing system

Reference:

- [Flickr Architecture](#)
- [Instagram Architecture](#).

Design a search engine

Reference:

- [How would you implement Google Search?](#)
- [Implementing Search Engines](#)

Design a recommendation system

Reference:

- [Hulu's Recommendation System](#)
- [Recommender Systems](#)

Design a tinyurl system

Reference:

- [System Design for Big Data-tinyurl](#)
- [URL Shortener API](#).

Design a garbage collection system

Reference:

- [Baby's First Garbage Collector](#).

Design a scalable web crawling system

Reference:

- [Design and Implementation of a High-Performance Distributed Web Crawler](#)

Design the Facebook chat function

Reference:

- [Erlang at Facebook](#)
- [Facebook Chat](#)

Design a trending topic system

Reference:

- [Implementing Real-Time Trending Topics With a Distributed Rolling Count Algorithm in Storm](#)
- [Early detection of Twitter trends explained](#)

Design a cache system

Reference:

- [Introduction to Memcached.](#)