Question 1 of 15

Which API protocol should you default to in system design interviews?

- GraphQL
- WebSockets
- gRPC

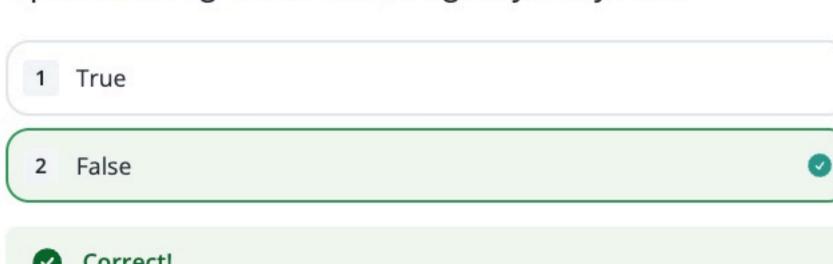


REST

REST should be your default choice as it's well-understood, has great tooling, and works for 90% of use cases. Only choose alternatives when you have specific requirements that REST can't meet.

Question 2 of 15

REST resources should represent actions like 'createUser' or 'updateBooking' rather than things in your system.



Correct!

REST resources should represent *things* in your system (users, bookings, events), not *actions*. Instead of 'createUser', you'd POST to '/users'. Instead of 'updateBooking', you'd PUT to '/bookings/{id}'.



When should you use path parameters vs query parameters in REST APIs?

Use path parameters when the value is required to identify the resource



- 2 Always use path parameters for better performance
- 3 Use them interchangeably
- 4 Always use query parameters for flexibility

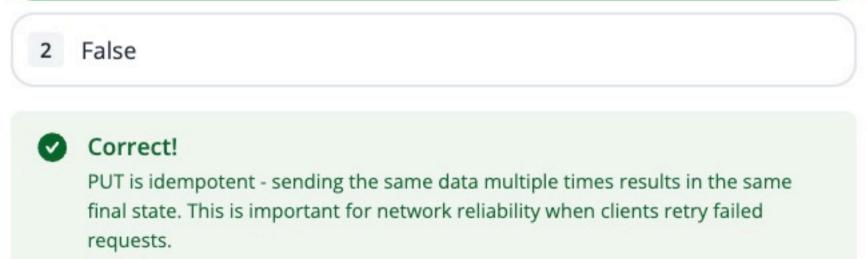
Correct!

Use path parameters when the value is required to identify the resource (e.g., '/events/{id}/tickets'). Use query parameters for optional filters or modifiers (e.g., '/events?city=NYC&date=2024-01-01').

Questi	on 4	of '	15

True

HTTP PUT requests are idempotent, meaning calling them multiple times produces the same result.



Question 5 of 15

Which pagination approach is more stable when new records

- are being added?
 - 1 Offset-based pagination
 - 2 Both are equally stable
 - 3 Neither handles new records well
 - 4 Cursor-based pagination

Correct!

Cursor-based pagination uses a pointer to a specific record instead of counting from the beginning, making it unaffected by new records being added during pagination.



Oue	stion	6 of 1
-----	-------	--------

When should you consider GraphQL over REST in system design interviews?

When you see clear over-fetching or under-fetching problems

Always, as GraphQL is more modern

When you need better performance

Never, REST is always sufficient

Correct!

Consider GraphQL when the interviewer mentions scenarios like different data needs for mobile vs web apps, or when you need to avoid over-fetching and under-fetching problems.















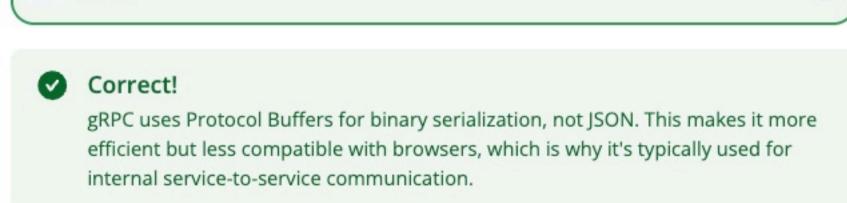








False



Question 8 of 15	F
What is the primary advantage of Protocol Buffers	over JSON?

- 1 Better browser support
- 2 Easier debugging
- 3 Smaller size and faster parsing
- 4 Better human readability



Protocol Buffers use binary encoding that results in smaller message sizes and faster parsing compared to JSON's text-based format, making them ideal for high-performance service communication.

Authentication verifies identity while authorization verifies permissions.

1 True



False

Correct!

Authentication proves the user is who they claim to be (identity verification), while authorization checks if that authenticated user is allowed to perform the specific

action they're requesting (permission verification).

Question 10 of 15

When should you use API keys vs JWT tokens?

- 1 Always use API keys for better security
 - Use JWT for user sessions, API keys for service-to-service communication
- 3 They are interchangeable
- 4 Use API keys for user sessions, JWT for services
- Incorrect.

Use JWT tokens for user sessions in web/mobile applications as they can carry user context and be stateless. Use API keys for internal service communication and external developer access.





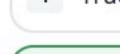




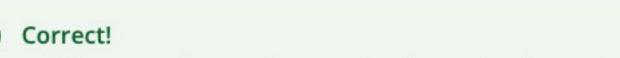


Question 11 of 15	
WT tokens require a database lookup to verify user	
nformation.	









JWT tokens encode user information directly into the token itself and are signed with a secret key. You can verify authenticity and read user info without database lookups, making them stateless.







4 401 Unauthorized



429 Too Many Requests is the appropriate status code when a client exceeds rate limits. This clearly indicates the nature of the problem and allows clients to implement appropriate backoff strategies.

Question 13 of 15
In RBAC (Role-Based Access Control), how are permissions

typically assigned?

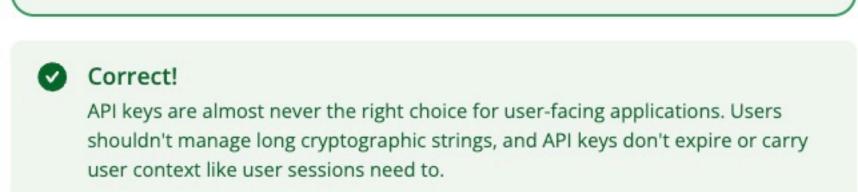
- 1 To API endpoints directly
- 2 To roles, which are then assigned to users
- 3 Directly to individual users
- 4 Through database queries

Correct!

In RBAC, permissions are assigned to roles (like 'customer', 'admin'), and then users are assigned to those roles. This simplifies permission management and ensures consistent access control.



False



What should you focus on most during the API design section of a system design interview?

Demonstrating solid engineering judgment with reasonable API choices



- 2 Using the most cutting-edge protocols
- 3 Perfect API specification with all edge cases
- 4 Memorizing every HTTP status code

Correct!

API design in interviews is about demonstrating solid engineering judgment, not creating perfect specifications. Focus on choosing the right protocol, modeling resources clearly, and showing understanding of basics like authentication and security.