Google Cloud

Partner Certification Academy

Professional Cloud Developer

v2309

Quiz questions*

Firebase Authorization

** These are for practice only and are not actual exam questions*

Question: In Firestore Security Rules, what declaration scopes the rules specifically to Firestore, preventing conflicts with other products?

- A. service cloud.firestore
- B. service cloud.database
- C. service firebase.firestore
- D. service google.firestore

Question: In Firestore Security Rules, which declaration specifies that rules should match any Firestore database in the project?

- A. match /databases/{database}/collections
- B. match /databases/{database}/documents

- C. match /firestore/{database}/documents
- D. match /cloud/{database}/documents

Question: What does the wildcard {city} in the match statement match /cities/{city} represent in Firestore Security Rules?

- A. Any city name in the 'cities' collection.
- B. A specific city name in the 'cities' collection.
- C. Any collection name under 'cities'.
- D. The name of the 'cities' collection.

Question: In Firestore Security Rules, how can you break down read and write into more granular operations?

- A. get, list, create, update, and delete
- B. read, write, create, and delete
- C. get, set, push, and remove
- D. read, write, push, and pull

Question: In Firestore Security Rules, if you want rules to apply to an arbitrarily deep hierarchy, which recursive wildcard syntax should you use?

- A. {name=*}
- B. {name=**}
- C. {name=...}
- D. {name=++}

Question: In version 2 of Firestore Security Rules, how many recursive wildcards can you have per match statement?

- A. One
- B. Two
- C. Three
- D. Unlimited

Question: In Firestore Security Rules, what happens if a document matches more than one match statement?

- A. Access is denied.
- B. Access is allowed if any of the conditions is true.
- C. Access is allowed only if all conditions are true.
- D. The first matching rule is applied.

Answers to Quiz questions

Firebase Authorization

Question: In Firestore Security Rules, what declaration scopes the rules specifically to Firestore, preventing conflicts with other products?

- A. service cloud.firestore
- B. service cloud.database
- C. service firebase.firestore
- D. service google.firestore

Correct Answer: A. service cloud.firestore

Explanation: The service cloud.firestore declaration is used to scope the security rules specifically to Firestore. This ensures that there are no conflicts between Firestore Security Rules and rules for other Google Cloud products.
Resource: Structuring security rules | Firestore | Google Cloud

Question: In Firestore Security Rules, which declaration specifies that rules should match any Firestore database in the project?

- A. match /databases/{database}/collections
- B. match /databases/{database}/documents
- C. match /firestore/{database}/documents
- D. match /cloud/{database}/documents

Correct Answer: B. match /databases/{database}/documents

Explanation: The match /databases/{database}/documents declaration is used to specify that the rules should match any Firestore database in the project.
Resource: Structuring security rules | Firestore | Google Cloud

Question: What does the wildcard {city} in the match statement match /cities/{city} represent in Firestore Security Rules?

- A. Any city name in the 'cities' collection.
- B. A specific city name in the 'cities' collection.
- C. Any collection name under 'cities'.
- D. The name of the 'cities' collection.

Correct Answer: A. Any city name in the 'cities' collection.

Explanation: The {city} wildcard syntax in the match statement means the rule applies to any document in the 'cities' collection, such as /cities/SF or /cities/NYC.
Resource: Structuring security rules | Firestore | Google Cloud

Question: In Firestore Security Rules, how can you break down read and write into more granular operations?

- A. get, list, create, update, and delete
- B. read, write, create, and delete
- C. get, set, push, and remove
- D. read, write, push, and pull

Correct Answer: A. get, list, create, update, and delete

Explanation: In Firestore Security Rules, read can be broken down into get and list, while write can be divided into create, update, and delete for more granular control.
Resource: Structuring security rules | Firestore | Google Cloud

Question: In Firestore Security Rules, if you want rules to apply to an arbitrarily deep hierarchy, which recursive wildcard syntax should you use?

- A. {name=*}
- B. {name=**}
- C. {name=...}
- D. {name=++}

Correct Answer: B. {name=**}

Explanation: The recursive wildcard syntax {name=**} is used in Firestore Security Rules to apply rules to an arbitrarily deep hierarchy.
Resource: Structuring security rules | Firestore | Google Cloud

Question: In version 2 of Firestore Security Rules, how many recursive wildcards can you have per match statement?

- A. One

- B. Two
- C. Three
- D. Unlimited

Correct Answer: A. One

Explanation: In version 2 of the Firestore Security Rules, you can have at most one recursive wildcard per match statement.
Resource: Structuring security rules | Firestore | Google Cloud

Question: In Firestore Security Rules, what happens if a document matches more than one match statement?

- A. Access is denied.
- B. Access is allowed if any of the conditions is true.
- C. Access is allowed only if all conditions are true.
- D. The first matching rule is applied.

Correct Answer: B. Access is allowed if any of the conditions is true.

Explanation: If multiple match expressions match a request in Firestore Security Rules, the access is allowed if any of the conditions is true.
Resource: Structuring security rules | Firestore | Google Cloud