
Firebase
RULES

Vi måste **ALLTID** validera både i frontend samt backend

I gränsnittet för att ge feedback till användaren

I databasen så att inte felaktig information lagras

Public

```
{
  "rules": {
    ".read": "true",
    ".write": "true"
  }
}
```

Vem som helst får skriva och läsa från hela vår database

Private

```
{
  "rules": {
    ".read": "false",
    ".write": "false"
  }
}
```

Ingen får skriva och läsa från hela vår database

Default

```
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

Inloggade användare får skriva till hela databasen

```
{
  "rules": {
    "users" : {
      ".read" : "auth != null",
      ".write" : "auth != null"
    },
    "todos": {
      ".read" : "true",
      ".write" : "auth != null"
    }
  }
}
```

Inloggade får skriva/läsa till `users`

Alla får läsa men inte skriva till `todos`

User

```
{
  "rules": {
    "users": {
      "$uid": {
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

`$uid` syftar på **key**

`auth.uid` syftar alltid på den inloggade användaren

read/write cascade

om en förälder har det så har barnen det

Men ett barn kan inte skriva över sin förälders rättigheter

Finns `true` på föräldern så spelar det ingen roll om barnet har `false`

4 regler

- `.read`
- `.write`
- `.validate`
- `.indexOn`

`.indexOn`

För snabbare sökning används **indexering**

Indexera det du söker mest på

Hjälper firebase att ta fram informationen snabbare

```
{
  "todos" : {
    ".indexOn" : "createdAt"
  }
}
```

Säg åt databasen vilka egenskaper du oftast kommer att sortera på

Gäller t.ex. `.orderByChild()`

Simulator finns under

`/Database/Rules`

".VALIDATE"

Speciella variabler i "rules"

- `$`
  - key
  - uid
  - postId
- `auth`
  - All information om nuvarande användare

Speciella variabler i "rules"

- `data`
  - Existerande data
- `newData`
  - Inkommande data
- `now`
  - Tiden just nu

```
{
  "rules": {
    "tweets": {
      "content": {
        ".validate": "newData.val().length <= 140"
      }
    }
  }
}
```

Värdet skrivs bara om den nya datans längd är <= 140

```
{
  "rules": {
    "tweets": {
      "content": {
        ".validate": "newData.val().length <= 140
                      && newData.val().isString()"
      }
    }
  }
}
```

Värdet skrivs bara om den nya datans längd är <= 140 och är en sträng

```
{
  "rules": {
    "tweets": {
      ".read" : "true",
      ".write" : "auth != null"
      "content": {
        ".validate": "newData.val().length <= 140
                      && newData.val().isString()"
      }
    }
  }
}
```

```
{
  "rules": {
    "tweets": {
      "$tweetId" : {
        ".write": "data.child(auth.uid).exists()"
        "content": {
          ".validate": "newData.val().length <= 140
                        && newData.val().isString()"
        }
      }
    }
  }
}
```

Om tweeten har ett barn som är mitt användarID får användaren skrivaccess

```
{
  "rules": {
    "tweets": {
      "$tweetId" : {
        ".validate": "newData.hasChildren(['content'])"
      }
  }
}
```

tweeten måste ha ett innehåll

LÄS MER

https://firebase.google.com/docs/database/security/