

Web Sockets - Firebase Database Real Time

@carlosmolanos





Powered by:



BOOT CAMP



CARRERA FULLSTACK JAVASCRIPT

Certifica CreaTIC con el apoyo de Platzi

● **Inicia:** 11 de agosto
Finaliza: 27 de Octubre

● **Lugar**
Laboratorio de la Corporación
Cluster CreaTIC

● **Horarios**
Lunes y viernes: 6:00 pm - 10:00 pm
Sábados : 8:00 am - 12:00 m y
2:00 pm - 6:00 pm

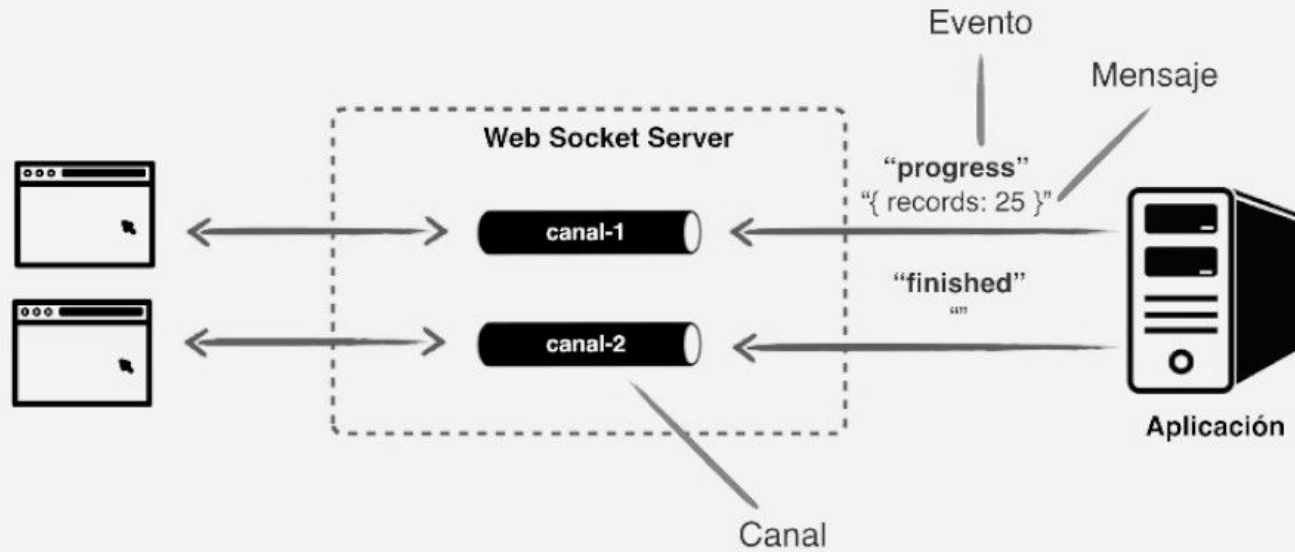


Que es WebSockets



WebSockets es una tecnología avanzada que hace posible abrir una sesión de comunicación interactiva entre el navegador del usuario y un servidor. Con esta API, puede enviar mensajes a un servidor y recibir respuestas controladas por eventos sin tener que consultar al servidor para una respuesta.

Arquitectura



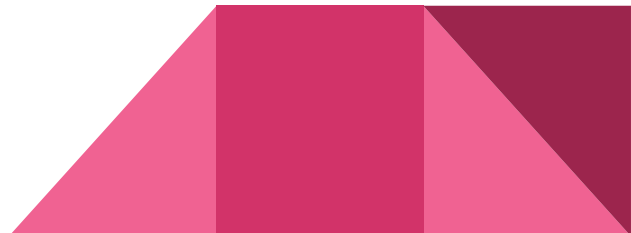
Firebase



Firebase

Firebase

Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando compilas apps multiplataforma con nuestros SDK de iOS, Android y JavaScript, todos tus clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.



Instalación y configuración

```
var config = {  
    apiKey: "apiKey",  
    authDomain: "projectId.firebaseio.com",  
    databaseURL: "https://databaseName.firebaseio.com",  
    storageBucket: "bucket.appspot.com"  
};  
  
firebase.initializeApp(config);  
  
// Get a reference to the database service  
var database = firebase.database();
```


Instalación y configuración

```
// Conectarse a la base de datos y a una “tabla” o colección de  
datos
```

```
var basedeDatos = firebase.database().ref('mensajes');
```

Eventos

Evento	Uso normal
child_added	Recupera listas de elementos o detecta elementos agregados a una lista. Este evento se activa una vez por cada elemento secundario existente y otra vez cuando se agrega un elemento secundario nuevo a la ruta de acceso especificada. El agente de escucha recibe una instantánea que contiene los datos del nuevo elemento secundario.
child_changed	Detecta cambios en los elementos de una lista. Este evento se activa cuando se modifica un nodo secundario. Esto incluye cualquier modificación en los descendientes del nodo secundario. La instantánea que se pasa al agente de escucha de eventos contiene los datos actualizados del elemento secundario.
child_removed	Detecta cuando se quitan elementos de una lista. Este evento se activa cuando se quita un elemento secundario inmediato. La instantánea que se pasa al bloque de devoluciones de llamada contiene los datos del elemento secundario que se quitó.
child_moved	Detecta cambios en el orden de los elementos de una lista ordenada. Los eventos <code>child_moved</code> siempre siguen al evento <code>child_changed</code> que generó el cambio en el orden de los elementos (en función del método de ordenamiento actual).



chat

sockets

Evento child_added

```
basedeDatos.on('child_added', function (data) {  
    $('#mensajes').append(`</p><p>${data.val().body}<p/>`);  
});
```

Enviar mensajes

```
$('#enviar').on('click', enviarMSG);  
$('#textoMsg').on('keypress',function(e){  
    if(e.which == 13) {  
        enviarMSG();  
    }  
});  
  
function enviarMSG(){  
    console.log('Enviar mensaje');  
    basedeDatos.push().set({body: $('#textoMsg').val()})  
    $('#textoMsg').val('');  
}
```

CRUD



Firebase

Create

```
$('#crearEmpresa').on('click', function () {  
  var empresa = {  
    nombre : "Creazion",  
    email: "info@creazionsoftware.com",  
    web: "www.creazionsoftware.com",  
    servicios: [{nombre: "web"},  
    {nombre: "movil"}]};  
  
    basedeDatos.push(empresa).then(function () {  
      console.log('Subio los cambios');  
    }).catch(function () {  
  
      console.log('error❏');  
  
    });  
  })
```

Read

```
basedeDatos.child("-LNhupSWlF8rWfW9z686").once('value',function (data) {  
    console.log(data.val());  
})
```

```
basedeDatos.limitToFirst(10).once('value',function (data) {  
    dataset = data.val();  
    console.log(data.val());  
    var empresas = $.map(data.val(),function (emp) {  
        console.log(emp);  
        return emp;  
    })  
    console.log(empresas);  
});
```


Update

```
$('#updateEmpresa').on('click', function () {  
    var empresa = {  
        nombre : "Creazion Cloud",  
        email: "info@creazionsoftware.com",  
        web: "www.creazionsoftware.com",  
        servicios: [{nombre: "games"},  
                    {nombre: "web"}]  
    };  
  
    basedeDatos.child("-LNhcp2lQN6p48RNIehf").update(empresa)  
    .then(function () {  
        console.log('Elimino el objeto');  
    })  
    .catch(function () {  
        console.log('Ocurrio un error');  
    });  
})
```

Delete (remove)

```
$('#deleteEmpresa').on('click', function () {  
    basedeDatos.child("-LNhcp2lQN6p48RNIehf").remove()  
    .then(function () {  
        console.log('Elimino el objeto');  
    })  
    .catch(function () {  
        console.log('Ocurrio un error');  
    });  
})
```

Filtros

```
basedeDatos.orderByChild('nombre')  
  .startAt('Cre').endAt('Cre\uf8ff')  
  .once('value', function (datos) {  
    console.log(datos.val());  
  })
```

Referencias

Developer Mozilla

<https://developer.mozilla.org/>

Otros:

<https://firebase.google.com>

