

Artículo

[Ricardo Paiva](#) · 22 abr, 2022 · Lectura de 3 min

[Open Exchange](#)

Cómo actualizar el Recurso Paciente usando fhir.js

¡Hola Comunidad!

Compartí mi experiencia trabajando por primera vez con FHIR en [este artículo](#).

En esa publicación, escribí cómo exploré los recursos FHIR y comenté la información que encontré útil en la documentación FHIR.

La primera versión de mi app solo muestra la información que ofrecen los Recursos FHIR.

Quería hacer que el usuario pudiera actualizar los detalles del paciente, así que busqué un ejemplo de cómo hacerlo.

La [página Github de fhir.js](#) tiene un ejemplo de cómo actualizar un recurso.

Update Resource

To update a resource, call `fhir.update({type: resourceType, id: identifier, resource: resourceObject})`. In case of success, the callback function will be invoked.

Example:

```
this.fhirClient.update({
  type: "Patient",
  id: 1,
  resource: {
    name: 'New Name'
  }
}).catch(function(e){
  console.log('An error happened while updating patient: \n' + JSON.stringify(e));
  throw e;
}).then(function(bundle){
  console.log('Updating patient succeeded');
  return bundle;
});
```

Viendo este ejemplo me dí cuenta de que era fácil.

Para actualizar el recurso, solo necesitas informar de las propiedades que quieres cambiar. ¡Guau!



Pensé que sería algo así:

```
this.fhirClient.update({
  type: "Patient",
  id: 1,
  resource: {
    name: 'Henrique',
```

```
        birthDate: '1984-02-10'
      }
    }).catch(function(e){
      console.log('An error happened while updating patient: \n' + JSON.stringify(e));
      throw e;
    }).then(function(bundle){
      console.log('Updating patient succeeded');
      return bundle;
    });
```

Pero no funcionó exactamente de esa manera.

Para que funcione, tenía que hacer algunos ajustes en mi código.

Estoy definiendo una variable con el objeto paciente que vuelve cuando relleno el formulario.

```
// Perform a search to retrieve patient details for a specific patient
window.loadForm = function (patientId) {
  client.search({
    type: 'Patient',
    query: {
      _id: patientId
    }
  }).then((res) => {
    const bundle = res.data;
    bundle.entry.forEach((patient) => {
      objPatient = patient;
    });
  });
}
```

Aquí la variable objPatient será el objeto que enviaré en el FHIR Update.

Cuando se hace clic en el botón Update, desencadena una función updatePatient.

Y la función updatePatient es responsable de actualizar el dato objPatient y después enviar el objeto al servidor FHIR.

```
window.updatePatient = function (patientId) {
  objPatient.resource.id = $("#fhirId").val();
  objPatient.resource.identifier[2].value = $("#SSN").val();
  objPatient.resource.name[0].given[0] = $("#firstName").val();
  objPatient.resource.name[0].family = $("#lastName").val();
  objPatient.resource.birthDate = $("#dateofbirth").val();
  objPatient.resource.gender = $("#gender").val();
  objPatient.resource.address[0].line[0] = $("#address").val();
  objPatient.resource.address[0].city = $("#city").val();
  objPatient.resource.address[0].state = $("#state").val();
  objPatient.resource.address[0].country = $("#country").val();
  client.update({
    type: "Patient",
    id: parseInt(patientId),
    resource: objPatient.resource
  }).catch(function (e) {
    showToast(1);
    $("#updateData").prop('disabled', false);
    throw e;
  }).then(function (bundle) {
    showToast(0);
  });
}
```

```
        $("#updateData").prop('disabled', false);  
        return bundle;  
    });  
};
```

Haciéndolo así, el servidor FHIR se actualiza con la información que necesitas.

Espero que estas piezas de código puedan ser útiles a alguien, algún día. 😊

Aquí puedes probar esta funcionalidad:

<http://iris-fhir-portal.eastus.cloudapp.azure.com:32783/csp/user/fhirUI/patientlist.html>

[#Consejos y trucos](#) [#FHIR](#) [#InterSystems IRIS for Health](#)

[Ir a la aplicación en InterSystems Open Exchange](#)

URL de
fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-actualizar-el-recurso-paciente-usando-fhirjs>