

Tutorial - Building a JavaScript App

Getting started

The SMART on FHIR JavaScript client library helps you build browser-based SMART apps that interact with a FHIR REST API server. It can help your app get authorization tokens, provide information about the user and patient record in context, and issue API calls to fetch clinical data.

To get started with the SMART on FHIR JavaScript client library, you'll need to:

1. Include a script tag

Include a `script` tag referencing the library. The latest code is always available *for download* (not live hosting) in GitHub at <https://raw.githubusercontent.com/smart-on-fhir/client-js/master/dist/fhir-client.js> (<https://raw.githubusercontent.com/smart-on-fhir/client-js/master/dist/fhir-client.js>).

You'll want to download and host this file alongside your app. Unless you're just prototyping – in which case you can use this rawgithub link (<https://raw.githubusercontent.com/smart-on-fhir/client-js/master/dist/fhir-client.js>) in your script tag's `src`. (Be sure never to use "rawgithub" in a deployment scenario, though!)

Note: if prototyping in JsFiddle (<https://jsfiddle.net>), then you will need to use the CDN hosted (<https://cdn.rawgit.com/smart-on-fhir/client-js/v0.1.8/dist/fhir-client.js>) version.

Including this script will create a global `FHIR` object for you to work with.

2. Create or obtain an instance of `FHIR.client`

In a typical workflow, you won't instantiate this object yourself – it will be created by `FHIR.oauth2.launch()`. But when you're prototyping, you can just create your own. By convention, we call this instance `smart`. So you can create a client via:

```
var smart = FHIR.client({
  serviceUrl: 'https://r2.smarthealthit.org',
  patientId: 'smart-1137192'
});
```

This object will be your touchpoint for making FHIR API calls. Your client will have some context built in, including:

- `smart.patient`
- `smart.user`

3. Use your client instance to execute FHIR API calls

You can make calls like:

- `read` : fetch a single resource, given its id
- `search` : search for resources that match a set of criteria

In general, you can make API calls that are scoped to the current patient by using `smart.patient`, as follows:

```
// Search for the current patient's conditions
smart.patient.api.search({type: 'Condition'});

// Search for the current patient's prescriptions
smart.patient.api.search({type: 'MedicationOrder'});
```

If you're writing a population-level app, you can query across patient records by using `smart.api`, as follows:

```
// Search for conditions added today
var todaysDiagnoses = smart.api.search({type: 'Condition', query: {dateRecorded: '2014-05-01'}});

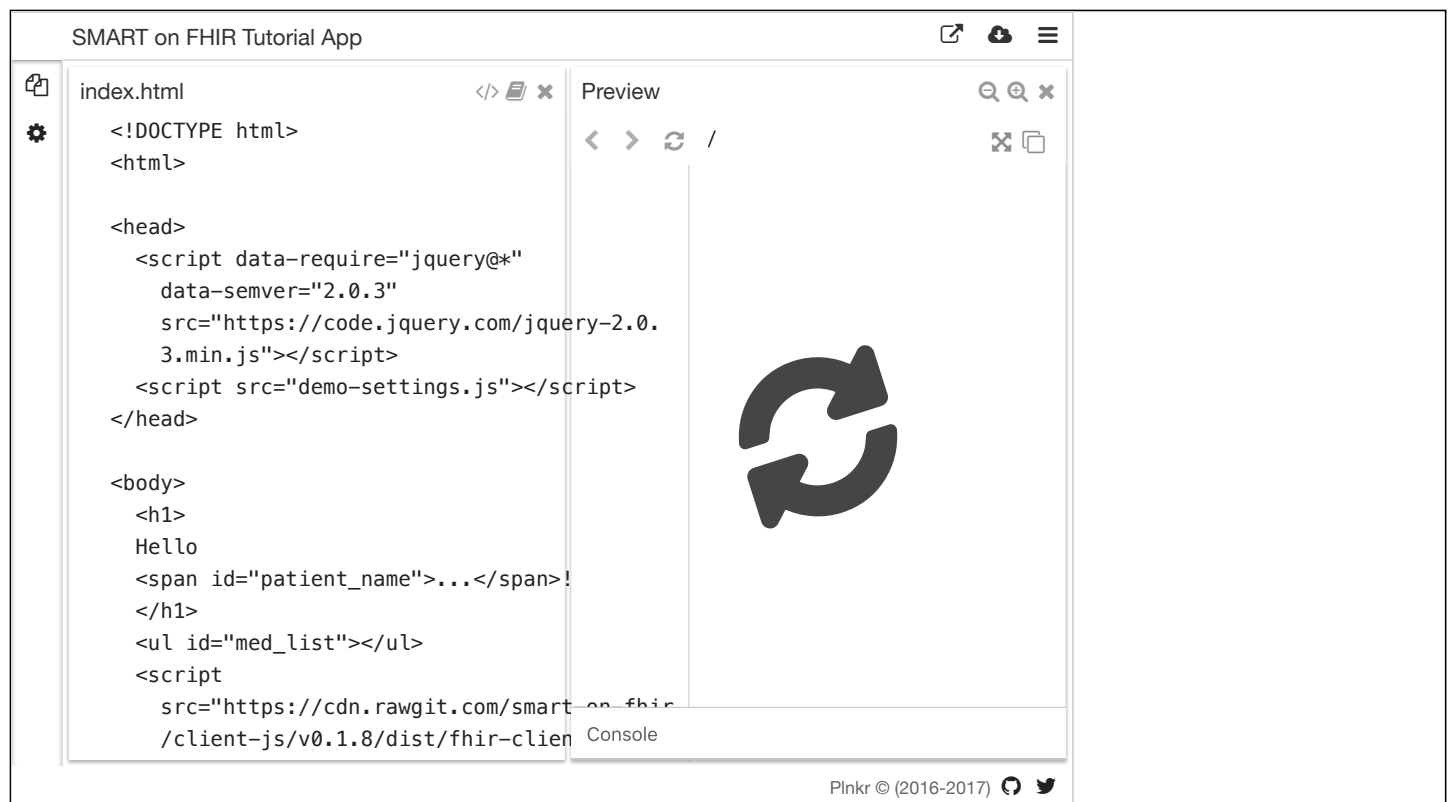
// Search for all statins prescribed today
var statinRx = smart.api.search({type: 'MedicationOrder', query: {dateWritten: '2014-05-01', name: 'statin'}});
```

These functions return `$.Deferred` objects, which you can work with simply by calling their `done` method:

```
statinRx.done(function(prescriptions){
  console.log(prescriptions);
});
```

In the `done` callback, you'll get a data structure containing search results.

Here's a complete example of the steps above:



For more details, see our JS client docs ([../clients/javascript](#))

The SMART on FHIR API is evolving in parallel with the [FHIR ballot releases \(http://hl7.org/fhir/timelines.html\)](http://hl7.org/fhir/timelines.html). If you spot problems, please [file an issue \(https://github.com/smart-on-fhir/smart-on-fhir.github.io/issues\)](https://github.com/smart-on-fhir/smart-on-fhir.github.io/issues). Or better yet, you can [edit this page \(https://github.com/smart-on-fhir/smart-on-fhir.github.io/edit/master/tutorials/javascript/index.md\)](https://github.com/smart-on-fhir/smart-on-fhir.github.io/edit/master/tutorials/javascript/index.md).