# Integrate your data with the Hyperledger Fabric blockchain

## Hyperledger Composer and REST APIs make data integration easy!

Paul O'Mahoney                                                                   July 13, 2017

When people think about integrating their data with a blockchain, such as a blockchain based on Hyperledger Fabric, they tend to have either a perplexed look or a complicated backstory. But it's not complicated. Hyperledger Composer provides easy integration, for the kind of data (such as assets, participants or transactions) and business network (such as a Commodities Trading, Digital Property, or Supply Chain network) that you work with. The easiest way to begin is by using REST APIs.

When people think about integrating their data with a blockchain, such as a blockchain based on Hyperledger Fabric, they usually have either a perplexed look or a complicated backstory.
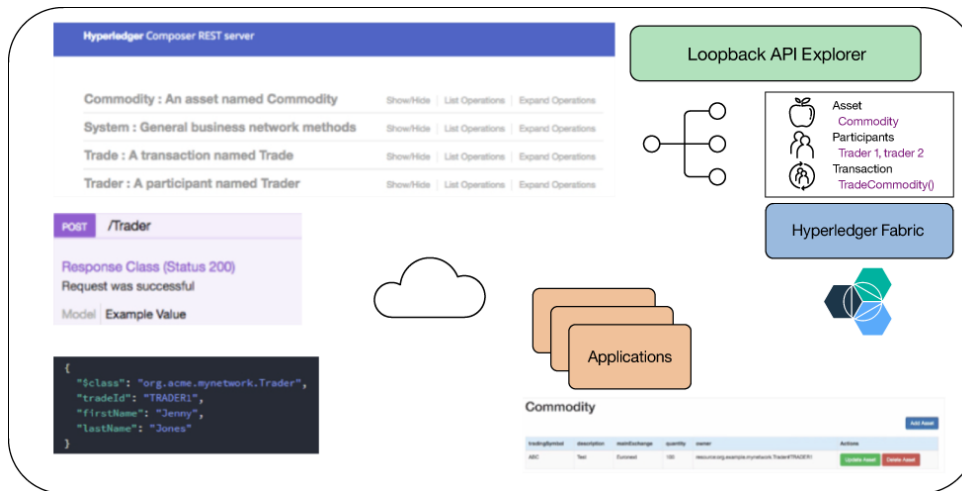
But data integration doesn't have to be complicated! The Hyperledger Composer application development toolset provides easy integration for your data (such as assets and participants that you create and update, and transactions that you execute) for the business network that you work with (such as a Commodities trading, Property, or Supply chain network).

> *" The simplest way to begin integrating your data is by using REST APIs. "*

## How it works

Once you deploy the definition of your business network created with Hyperledger Composer, you can instantly consume the REST APIs that Composer makes available. Under the covers, the Loopback Node.js framework is used to create the open APIs, the structure and format for which are defined by Swagger. Of course, the Composer Node SDK is available as well, but if you are creating a web application (or running in a non-Node.js environment), the REST API provides a handy layer of abstraction and separation.

## Figure 1. Components of the Hyperledger Composer REST server



When you install Hyperledger Composer in a development environment, the Hyperledger Composer REST Server is installed. Once you launch it from the command line, it asks one-time questions to discover your business network, which you've also defined through Composer. Once you answer those questions, Composer connects you to the running Hyperledger Fabric blockchain and discovers the REST APIs and generates their schemas for Loopback to import.

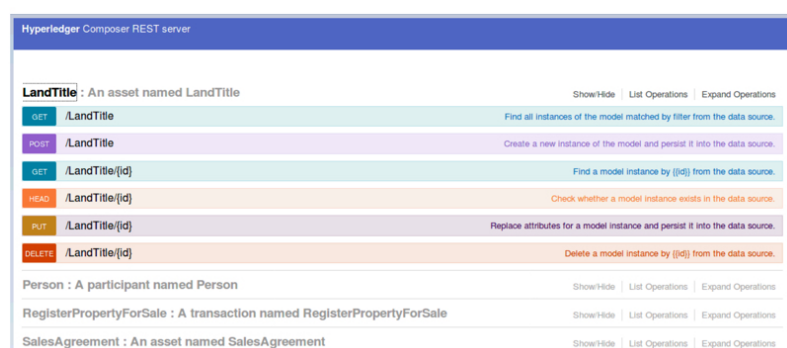## Figure 2. Hyperledger Composer makes the REST APIs available



The Composer REST server uses the Loopback API Explorer to both inspect and test the generated APIs. It's really that easy. The Loopback API Explorer is available through the browser, in this case at http://localhost:3000/explorer, as shown below. This is a sample Digital Property network previously deployed. Immediately, it shows the Asset, Participant, and Transactions types.

## Figure 3. Model-based APIs for a sample property network

# Adding your data

You can query an asset on the running blockchain by using the GET operation.

## Figure 4. Querying assets with GET



```
Hyperledger Composer REST server

LandTitle : An asset named LandTitle                          Show/Hide | List Operations | Expand Operations

GET    /LandTitle                                             Find all instances of the model matched by filter from the data source.

Curl

curl -X GET --header 'Accept: application/json' 'http://localhost:3000/api/LandTitle'

Request URL

http://localhost:3000/api/LandTitle

Response Body

[
  {
    "$class": "net.biz.digitalPropertyNetwork.LandTitle",
    "titleId": "LID:1148",
    "owner": "resource:net.biz.digitalPropertyNetwork.Person#PID:1234567890",
    "information": "A nice house in the country",
    "forSale": true
  },
  {
    "$class": "net.biz.digitalPropertyNetwork.LandTitle",
    "titleId": "LID:6789",
    "owner": "resource:net.biz.digitalPropertyNetwork.Person#PID:1234567890",
    "information": "A small flat in the city"
  }
]
```

Similarly, you can add an asset, add a transaction, and so on, by using the POST operation in the prescribed JSON format. You can do this for assets, participants, transactions as shown below. You simply supply the data in JSON format and use the POST API operation.

## Figure 5. Example of a Landtitle asset API POST



```
POST   /LandTitle
[
  {
    "$class": "net.biz.digitalPropertyNetwork.LandTitle",
    "titleId": "LID:1148",
    "owner": "resource:net.biz.digitalPropertyNetwork.Person#PID:1234567890",
    "information": "A nice farmhouse in Wisconsin",
    "forSale": true
  },
  {
    "$class": "net.biz.digitalPropertyNetwork.LandTitle",
    "titleId": "LID:6789",
    "owner": "resource:net.biz.digitalPropertyNetwork.Person#PID:1234567890",
    "information": "A small flat in the New York city"
  }
]
```

# Summary

If you combine this integration with the power of application generators, such as the yo Angular 2+ generator within the Hyperledger Composer suite, you can have a skeleton application project up and running in your browser in minutes and conduct testing between your running application (via Loopback and REST) and a runtime Hyperledger Fabric blockchain.

Complicated, it ain't!

Get a monthly roundup of the best free tools, training, and community resources to help you put blockchain to work.
**Current issue** | **Subscribe**

# Next steps

- Check out the Hyperledger Composer Developer Tutorial. It walks you through creating a functioning application for an end-to-end blockchain solution. You'll get a chance to try out the REST API integration for yourself.
- Read more about Hyperledger Composer in this in-depth introduction.
- Join the Hyperledger Composer community, ask a question on Stack Overflow, or chat on the Rocket.Chat discussion channels.

# Related topics

- [Get going quickly with blockchain using Hyperledger Composer](#)
- [Building a blockchain PoC in ten minutes using Hyperledger Composer](#)
- [IBM Blockchain Developer Center](#)
- [IBM Blockchain service on IBM Cloud](#)
- [Blockchain essentials (free course for developers)](#)