



# Tutorial A1: Introduction

---

Estimated time: **10 minutes**

In this tutorial you will:

- Learn what blockchain is and why it is important
- Learn about the Linux Foundation Hyperledger Project and Hyperledger Fabric
- Learn about IBM Blockchain Platform and the VS Code extension
- Familiarize yourself with the development environment

At the end of this tutorial you will be invited to gain the 🏆 [IBM Blockchain Essentials](#) badge.

As you read these tutorials, look out for things that you need to do. They are always written on their own line, are numbered sequentially, and look like the following:

**A1.1:** Expand the 'What is blockchain' section below to get started.

If you run into problems, first check back to make sure that you have completed all of these numbered steps in the correct order.

---

### ► What is blockchain?

The generally accepted definition of a blockchain is a *distributed ledger* with *smart contracts*.

Businesses generate wealth by flowing assets (goods and services) over business networks. Every time an asset moves, this is referred to as a *transaction*. Transactions are logged on a *ledger*, which is a record of the inputs and outputs of a business.

The problem with ledgers is that each one is owned by a single organization, which means that when a transaction involves multiple organizations, it could be recorded differently by each of them and ledgers can get out of sync. Ledgers therefore need to be *reconciled*, and when transactions don't tally disputes occur, the resolution of which can be slow and expensive.

Having a distributed ledger means that all participants of the business network see the same information. The ledger is not held in any one single place, which avoids the need for reconciliation and dispute resolution, and removes any single point of failure.

### Smart contracts

In the business world, transactions are underpinned by *contracts* which are the pre-requisites for the transaction. If I buy a car from you and it breaks down a month later, the terms of our contract might say that you are liable for the repair.

The problem with contracts is that they are often ambiguous. Deciding the meaning of a contract can again involve lengthy and costly dispute resolution processes.

Blockchain introduces *smart contracts*, which describe in code what a transaction stored on the shared ledger does. When I buy your car, the smart contract might describe the algorithm to check that you're the current owner of the car, that I have the means to pay and then if everything's OK, transfer the money to the seller and the car's ownership to me.

Each transaction is the request to call a smart contract with a given set of input parameters. In Hyperledger Fabric, which is the blockchain implementation that we will be using, smart contracts interact with a database called a *world state*. This is distinct from the transaction log; the world state might contain the list of current car owners for example, but the log would record the transaction that describes the request for me to buy your car.

## Qualities of service

Blockchain provides *cryptographic proof* of the transactions logged within it, and ensures each participant of the network sees the same information, using a process called *consensus*. Transactions cannot be subsequently amended or deleted.

This is a key differentiator of blockchain over a standard database: with databases, you have administrators who have control over the data stored within it. With shared databases you potentially have multiple administrators in multiple organizations, and there is no easy way to guarantee that the data has not been tampered with.

**Want to know more?** Download the free e-book: <http://ibm.biz/BlockchainOReilly>

## Blockchain for business

Blockchain is commonly associated with Bitcoin, the cryptocurrency and peer-to-peer payments system. Bitcoin uses a blockchain as the ledger to record its transactions and a resource-intensive method of consensus called *proof of work*. Bitcoin favors anonymity; even though the ledger is public, it's almost impossible to determine who is behind a bitcoin transaction.

Businesses often have a different operating environment. For example, they are required to carry out Know Your Customer (KYC) and Anti-Money Laundering (AML) checks, which require businesses to know who they are dealing with. This means that business blockchains require identifiable participants and favor features such as *privacy* and *confidentiality*. Knowing who's behind a transaction helps to remove incentives for fraud and as such, resource-intensive methods of consensus are not needed.

And of course, business ledgers also store more than cryptocurrency transactions; any asset of worth to a business can be transacted and logged on a business blockchain.

□ **A1.2:** Expand the 'Introduction to Hyperledger Fabric' section below to continue.

---

## ► Introduction to Hyperledger Fabric

In early 2016, the Linux Foundation formally announced the creation of the *Hyperledger* project, which aims to develop blockchain technologies that specifically cater for the requirements of businesses.

Hyperledger is not a single thing - it is a collection of cross-industry frameworks and tools that come under a single open license and governance structure.

Find out more about the Hyperledger Project at <https://www.hyperledger.org/>

One of the earliest and most widely implemented Hyperledger technologies is *Hyperledger Fabric*, which is an implementation of a shared, replicated ledger with support for smart contracts and pluggable frameworks for features such as privacy and consensus. It continues to be developed by hundreds of developers in the community representing dozens of different organizations.

Read the Hyperledger Fabric technical documentation at <https://hyperledger-fabric.readthedocs.io/en/>

Hyperledger Fabric underpins many of the world's most successful blockchain implementations, including Food Trust, TradeLens and we.trade.

## Hyperledger Fabric technology

A Hyperledger Fabric network consists of three key types of components:

- **Peer node:** holds a copy of the ledger and is responsible for running smart contracts.
- **Orderer node:** part of a distributed ordering service that agrees the order that transactions are added to the ledger.
- **Certificate Authority:** responsible for issuing the certificates that identify users and organizations on the network.

Each organization that joins a Hyperledger Fabric network will typically own peers and certificate authorities. There needs to be one ordering service in each network, although this can be distributed between organizations.

As we progress through these tutorials, we will work with each of these components.

□ **A1.3:** Expand the 'Introduction to IBM Blockchain Platform' section below to continue.

---

## ► Introduction to IBM Blockchain Platform

IBM Blockchain Platform is a set of tools and services that allow users to build and operate Hyperledger Fabric networks. Broadly, it consists of two main elements:

- **IBM Blockchain Platform Extension for Visual Studio Code.** This is the tool you are using now. It allows developers to create Hyperledger Fabric smart contracts and the applications that interact with them. Developers can test their applications using an embedded instance of Hyperledger Fabric, or connect to a remote Hyperledger Fabric network.
- **IBM Blockchain Platform Console.** This allows users to create and manage their Hyperledger Fabric components. The components can be run on IBM Cloud or on any OpenShift-based environment, including on-premises.

Learn more about the IBM Blockchain Platform here: <https://www.ibm.com/blockchain/platform>

In later tutorials you will learn more about the console. For now, you will start to familiarize yourself with the IBM Blockchain Platform developer tools.

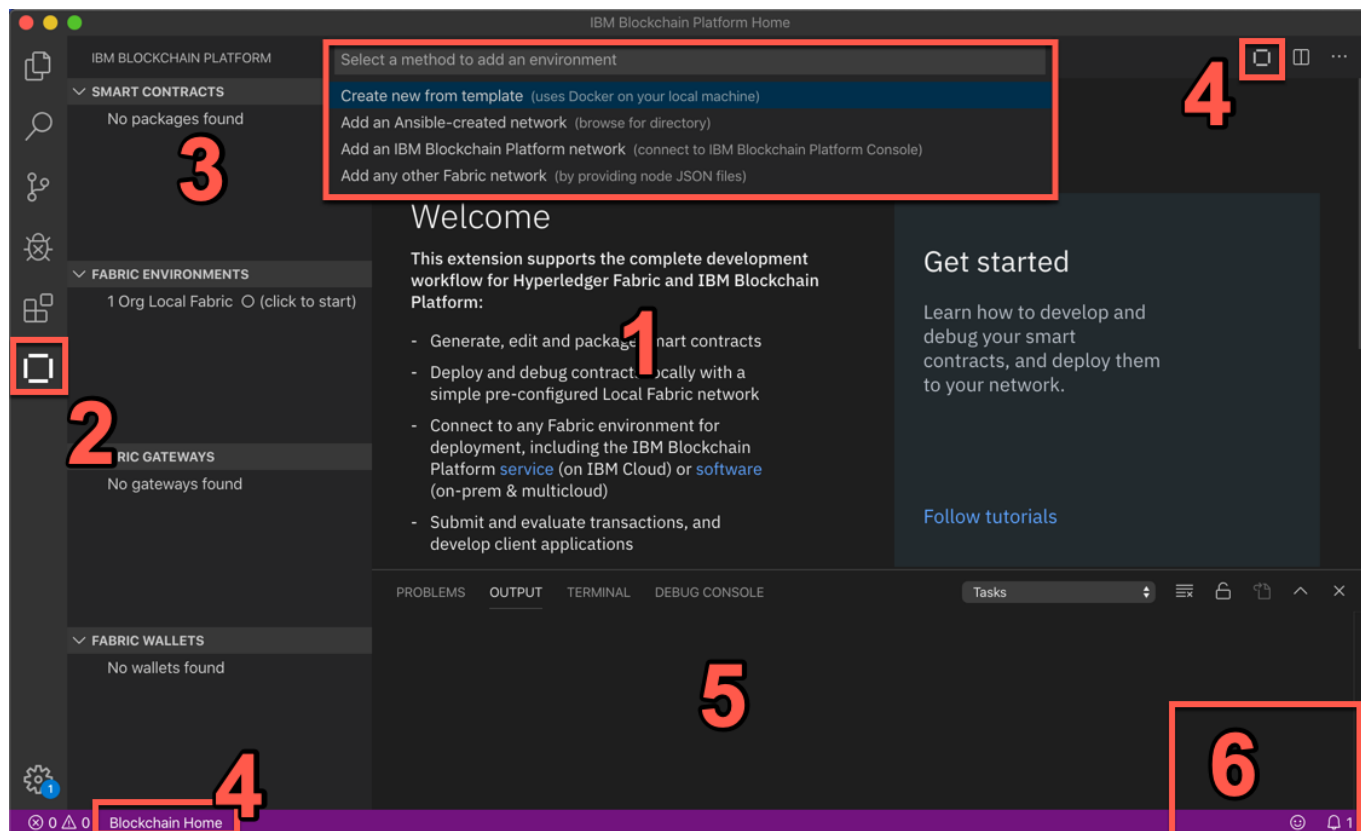
□ **A1.4:** Expand the next section to continue.

---

### ► IBM Blockchain Platform Extension for Visual Studio Code

From now on, you will be navigating the VS Code user interface, so begin by ensuring that you can comfortably see both the tutorial text and the VSCode editor. VS Code allows you to right-click an editor window and select 'Split' to arrange your open windows. Alternatively, select 'File' -> 'New Window' to create a second VS Code window into which you can load the tutorials.

We will start by understanding how to navigate VS Code. This is not intended to be a tutorial on VS Code, but the basic UI elements that are relevant to IBM Blockchain Platform are described in the following diagram:



1. **Editor pane:** Used for editing smart contracts and application code.
2. **The IBM Blockchain Platform icon in the VS Code sidebar:** Shows or hides the IBM Blockchain Platform sidebar.
3. **IBM Blockchain Platform sidebar:** A set of views that show the Hyperledger Fabric resources you are working with: smart contracts, environments, gateways and wallets.
4. **IBM Blockchain Platform home page:** At both the top and bottom of the screen are icons that allow you to return to the IBM Blockchain Platform welcome screen if you get lost.
5. **Output:** The results of running any transaction are displayed here. Also note the neighboring tabs that allow you, among other things, to run Terminal commands in an embedded shell.
6. **Notifications:** Timely information is displayed here, such as the results of deploying a new smart contract. If you click on the icon in the corner you can usually revisit any notifications you've missed.
7. **Command palette:** When you tell VS Code to do something, such as package a smart contract, VS Code will invite you to enter any parameters here. You can also bring up the command palette before you invoke any command by selecting "View -> Command Palette". (All of the IBM Blockchain Platform commands start with "IBM Blockchain Platform" so if you start typing this in the command palette you can quickly filter to the command you need.)

If you're not familiar with VS Code, it's worthwhile spending a few minutes clicking around to get yourself comfortable with it. When you are ready, continue on.

 **A1.5:** Expand the next section to continue.

## ► 🏆 IBM Blockchain Essentials

Blockchain solves a very specific problem in transactional networks: by providing cryptographic proof that information has been shared identically between participants, blockchain helps to encourage trust. From this

simple idea there is a significant opportunity to simplify business networks by removing friction related to disputes.

By completing this first tutorial, you will have gained the knowledge you need to pass the 🏆 [IBM Blockchain Essentials](#) course.

Clicking on the link above will take you to the course materials. Use these materials to consolidate what you've learned, then take the short quiz to demonstrate your understanding. If you are logged into [Acclaim](#) you will then be able to gain the badge.

In the next tutorial we will look at creating your first smart contract.

---

→ [A2: Creating a smart contract](#)