Pay Bonus By Blockchain(for developer)

Create Wallet

The command line tools can be obtained as a zipfile/tarball from the releases page of the project repository, under the Downloads section, or for OS X users via Homebrew, or for Arch linux users via the AUR.

more info.

Windows

Use PlwerShell input the commend.

$ ./web3j

To generate a new Ethereum wallet:
$ ./web3j wallet create

// then type the password
Please enter a wallet file password:
Please re-enter the password:
Please enter a destination directory location
[C:\Users\Godfather\AppData\Roaming\Ethereum\testnet\keystore]: keystore
Creating directory: keystore ... complete

To update the password for an existing wallet:

$ ./web3j wallet update <walletfile>

Get Wallet Address

Login [MyEtherWallet](http://www.myetherwallet.com) using your wallet file and passowrd, then you will find your address.

Deployment Project

Maven

Java 8

```xml
<dependency>
  <groupId>org.web3j</groupId>
  <artifactId>core</artifactId>
  <version>4.3.0</version>
</dependency>
```

Gradle

Java 8

```groovy
compile ('org.web3j:core:3.4.0')
```

The source code in [Github](https://github.com).
Signing Infura Service

The Infura service by ConsenSys, provides Ethereum clients running in the cloud, so you don’t have to run one yourself to work with Ethereum.

When you sign up to the service you are provided with a token you can use to connect to the relevant Ethereum network:

Main Ethereum Network:
https://mainnet.infura.io/<your-token>

Test Ethereum Network (Rinkeby):
https://rinkeby.infura.io/<your-token>

For example, my token is:
https://rinkeby.infura.io/v3/34ada3de9b9e4186b365975ba1843c4f

You can create the client just like the regular HTTPClient:

```java
Web3j web3 = Web3j.build(new
HttpService("https://rinkeby.infura.io/<your-token>"));
Web3ClientVersion web3ClientVersion = web3.web3ClientVersion().send();
System.out.println(web3ClientVersion.getWeb3ClientVersion());
```

Ethereum Testnets

There are a number of dedicated test networks in Ethereum, which are supported by various clients.

- Rinkeby (Geth only)
- Kovan (Parity only)
- Ropsten (Geth and Parity)

For development, its recommended you use the Rinkeby or Kovan test networks. This is because they use a Proof of Authority (PoA) consensus mechanism, ensuring transactions and blocks are created in a consistent and timely manner. The Ropsten testnet, although closest to the Mainnet as it uses Proof of Work (PoW) consensus, has been subject to attacks in the past and tends to be more problematic for developers.

You can request Ether for the Rinkeby testnet via the Rinkeby Crypto Faucet, available at https://www.rinkeby.io/.

Pay Bonus

I edited three transfer functions with different parameters.
public void transfer(String amount, String url, String pwd, String walletPath, String toAddress) throws Exception {
    Web3j web3j = Web3j.build(new HttpService(url));
    log.info("Connected to Ethereum client version: 
    + 
    " + web3j.web3ClientVersion().send().getWeb3ClientVersion());
    Credentials credentials = 
        WalletUtils.loadCredentials(pwd, walletPath);
    log.info("Credentials loaded");
    log.info("Sending "
        + Convert.fromWei(amount, Convert.Unit.ETHER).toPlainString() + ", Ether")
        BigDecimal num = new BigDecimal(amount);
        TransactionReceipt transferReceipt =
        Transfer.sendFunds(
            web3j, 
            credentials, 
            toAddress, // you can put any address here 
            num.multiply(BigDecimal.ONE), 
            Convert.Unit.WEI) // 1 wei = 10^-18 Ether .send();
        log.info("Transaction complete, view it at 

Check the Balance

You can check your balance in Rinkeby.
Just search your wallet address.